ADVANCES IN CONNECTIVITY-BASED POSITIONING FOR MOBILE WIRELESS SENSOR NETWORKS

STUART K. MACLEAN

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTORATE OF PHILOSOPHY

GRADUATE PROGRAM IN DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO
APRIL 2013

ADVANCES IN CONNECTIVITY-BASED POSITIONING FOR MOBILE WIRELESS SENSOR NETWORKS

by **Stuart K. MacLean**

a dissertation submitted to the Faculty of Graduate Studies of York University in partial fulfilment of the requirements for the degree of

DOCTORATE OF PHILOSOPHY

Abstract

A sensor network is a new area of application for computing devices which is developing as a consequence of the progression of the computing revolution. A network is composed of small, resource limited, devices that can take local measurements of phenomena such as temperature, pressure, and humidity. The devices communicate wirelessly.

The positions of sensor devices provide a spatial context for measured phenomena however the positions are not typically known after deployment. Therefore the estimation of each device's position is important. Device mobility makes position estimation especially challenging. In connectivity-based position estimation, the presence, or absence, of direct communication between devices constraints the possible positions of a device.

This dissertation investigates connectivity-based position estimation for systems of mobile devices. The product of this dissertation is a lower bound on the positional error, a new positioning algorithm, and the successful execution of the algorithm

on deployments of sensor devices.

The analysis of the positional error of a positioning algorithm for mobile devices is a challenging problem. A lower bound on the expected positional error incurred by any connectivity-based positioning algorithm is derived in this work. An analysis of the impact of past constraints on the lower bound reveals that the benefit of additional constraints from the past diminishes as the constraints age.

A distributed positioning algorithm called Orbit is proposed for stationary and mobile sensor networks. Orbit identifies network structures that are used to impose new constraints on device positions, which reduces positional error. The set of possible positions of a device may form isolated regions which is problematic. Orbit removes some of these isolated regions, which reduces positional error. The performance of Orbit and another recent algorithm are evaluated under different communication and mobility models. A performance analysis demonstrates that Orbit outperforms the other under a variety of parameters.

To verify the Orbit algorithm is amenable to sensor networks it is implemented on resource limited hardware. The result is an autonomous sensor network that is tested in several deployments. The position estimates from the sensor network are comparable to those from the simulation of Orbit.

Acknowledgements

I wish to express my gratitude to my supervisor, Professor Suprakash Datta, for his guidance and advice throughout my time at York University. He has spent endless hours working with me and these efforts have improved the quality of the dissertation.

Thanks to my dissertation committee members, Professors Nick Cercone, Eric Ruppert, Walter Whiteley, Sagar Naik, and Mokhtar Aboelaze, for their hard work in reading the dissertation and providing constructive feedback.

Finally, I would like to thank my family and friends for their patience and understanding. Their words of encouragement have been especially motivating.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The progression of the computing revolution is yielding ever shrinking miniature computing devices. The current form for these computing devices is a small circuit board containing a microcontroller unit (MCU). A MCU is a highly integrated circuit containing processors, memory, clocks and possibly other peripherals, like a radio or electromechanical components, on a single chip. The processors and memory are used for computation while other peripherals like the radio are used for wireless communication. While not all components need to be integrated directly on the MCU and may instead be spread out on a circuit board, this type of design can reduce the total physical size of the device.

A microcontroller can be coupled with a sensing device whose role is to quantify a phenomenon by taking measurements. The phenomena that can be observed by a sensor are numerous and include temperature, pressure, vibration, moisture, and even quantities like angular velocity and acceleration of the sensor device. If a microcontroller is powered with a battery, then systems of wirelessly communicating sensor devices can be deployed in large regions over long periods of time. Such a system provides a means of analyzing an environment via many local observations made by the sensor devices.

A system of communicating sensor devices is called a wireless sensor network. These networks can be deployed with large numbers of sensors in an ad hoc manner. The deployment of sensor devices is the process of embedding the sensors in an environment. Improving the coverage of an environment by increasing the number of sensor devices generally increases the measurement accuracy.

1.1.1 Desirable Attributes of a Wireless Sensor Device

There is no precise definition of a sensor device but there are some necessary attributes which are important features of the system design. A sensor device is physically small in size, capable of wireless communication and has a battery as the primary power source.

Size: A sensor device is expected to occupy a small volume of space since it

may be attached to another object that must be able to bear its weight. The device should also be unobtrusive to its environment.

Communication: Traditionally networks of computers have relied on an existing infrastructure for power and networking services like message routing. The deployment of sensor networks occurs in environments where there is no preexisting infrastructure. This results in an ad hoc organization of the sensor devices which places restrictions on the use of wired connections. A sensor device must be capable of wireless communication so that sensor measurements can be shared across the sensor network. A sensor device is expected to be powered by a battery which is small in capacity due to size restrictions. Integrated energy harvesting mechanisms like solar cells have been proposed for inclusion in sensor devices for battery charging but there are still limits on the rate of consumption of energy. The use of wireless technology gives a great deal of flexibility in the configuration of a sensor network but also imposes extreme limitations on the processing and memory resources available for computation.

Cost: Another consideration in sensor network design is monetary cost. The cost is difficult to quantify because it depends on changing market conditions. What can be said is that proposed sensor networks consist of large numbers of sensor devices so the unit cost of sensor devices must be low compared to the total system cost. This restriction will impact the design of the sensor devices in a way

that limits the available resources of an individual sensor device.

The aforementioned attributes: small size, wireless communication and battery power, make sensor devices suitable for mobile scenarios. In these cases a device is attached to an object whose position changes over time. The dynamics of movement are dependent on the scenario and may not be discernible by the device.

1.1.2 Examples of Wireless Sensor Devices

To give the reader some examples of the limited computational resources available to sensor devices, some devices commonly used for sensor network applications are described.

The MICA2 mote by Crossbow Technology is an early generation sensor device released in 2002 [41]. This device contains an ATMEGA123L CPU running at a clock rate of 8 Mhz and contains 4 kB of random access memory. The radio used by the device is contained on a separate chip and is attached to a whip antenna. Including two AAA batteries, the dimensions of the device are 58 x 32 x 7 mm and its weight is 63.82g. By comparison, the first personal computer created by Apple in 1976, the Apple I, had a MOS 6502 CPU with a clock running at 1 MHz and 4kB of random access memory.

The EZ430-RF2500 mote by Texas Instruments is another popular sensor device used for sensor networks that was released in 2007 [24]. This device contains an

MSP430 CPU running at a clock rate of 16 Mhz and contains 1kB of random access memory. The device contains a radio that is located on a separate chip. Including two AAA batteries, the dimensions of the device are 92 x 23 x 12 mm and its weight is 30.5g.

A more recent sensor device called the ECO mote has been designed for medical applications and was released in 2008 [50, 49]. The device contains a Nordic nRF24E1 MCU running at 16 MHZ and has 4 kB of random access memory. The MCU has an integrated radio which is connected to a PCB antenna (an antenna that is implemented with a conductive trace on the circuit board). The dimensions of the device, including the custom lithium-ion polymer battery, are 13 x 11 x 7 mm and its weight is 1.8g, At a volume of approximately 1 cm^3, the reduction in size is significant when compared with other solutions and it is an excellent demonstration of a device that is approaching the idealized physical form of a sensor device. The highly integrated design is partly responsible for much of the reduction in size. The other factor is the physical size of the battery which is smaller than the AAA batteries commonly used by sensor devices. With just 34 mAh capacity, this battery is far smaller capacity than a standard AAA battery which typically has a 1000–1200 mAh capacity.

1.1.3 Applications of Wireless Sensor Networks

There are many proposed applications for sensor networks that take advantage of their features. The application areas are diverse, ranging from agricultural and infrastructure monitoring, to wildlife and asset tracking. Applications for sensor networks have also been identified in the domains of medicine and the military.

A commonly proposed application of sensor networks is forest fire detection. The current methods for detecting forest fires include satellite imaging and direct detection via human beings stationed in watch towers. The drawback of using satellite imagery is delayed detection and cloud interference. The drawback of forest fire detection by humans is the monetary cost. A sensor network that is deployed by dropping sensors from aircraft could provide a cost effective approach to fire detection using temperature, humidity and smoke sensing devices [20].

The intensive research on sensor networks over the past decade is beginning to materialize into real world applications. In Spain, researchers have designed and implemented a sensor network composed of different types of sensor devices for the purpose of precision horticulture [33]. In this application, a set of sensors were deployed in a field containing crops of broccoli to measure soil characteristics like temperature, moisture and salinity. These measurements of environmental conditions were used to study crop development. The sensor network provided

real-time measurements which were used to ensure conditions were favourable for crop development.

Some researchers have proposed the use of sensor networks to study the behaviour of animals by tracking their positions [7, 32, 18]. LynxNet is a recently proposed system for tracking animals equipped with sensor devices attached to a collar [69]. This system forms a network composed of a mixture of mobile and stationary sensors and is designed to track the migration of the Eurasian lynx. A sensor device captures either (i) rotation and acceleration data or, (ii) position, temperature and ambient light data. These measurements are periodically sent back to a base station for analysis. The position of the sensor device is acquired via the global positioning system. Interestingly the daily energy consumption of the GPS component is 87.52 mW which is approximately 95% of the total daily consumption. According to the authors, this figure can be improved with a different GPS unit. Nevertheless, it reveals the energy consumption of a GPS unit is disproportionately high compared to the energy consumption of other components of the sensor device.

Although these sensor solutions are preliminary, they show progress toward fruitful applications of sensor networks in the real world.

1.1.4 Determination of the Positions of Sensor Devices

A prerequisite for many proposed applications of sensor networks is the knowledge of the positions of sensor devices. The positional information provides a spatial context for the phenomenon observed by the sensor devices. The positions of the sensor devices also provides support for some network operations like routing and network maintenance. A geometric routing protocol makes use of sensor positions for effective message delivery between sensors [27]. The intended longevity of a some sensor networks demands that the state of sensor components like batteries need to be monitored and managed accordingly [38]. The knowledge of sensor positions can assist in this process.

In many scenarios, the position of the device in its environment is not known a priori so position estimation is critical to the application. Therefore, a mechanism is required for the determination of the position of sensor devices after deployment. The estimation of the position of sensor devices is known as *positioning* (or localization) and is the central topic of this dissertation.

1.2 The Positioning Problem for Sensor Networks

This dissertation considers sensor networks that are collections of communicating sensor devices. The position of each device is to be determined. Each device consists

of a CPU and memory for computation, and a radio for wireless communication. The devices, herein referred to as **nodes**, are located in some environment and they are able to wirelessly communicate with other devices by sending and receiving messages. The communication region of a device is the region of space where a message sent by one node can be received by other nodes. Figure 1.1 illustrates radio connectivity between nodes. The organization of the nodes is ad hoc in the sense that there is no pre-existing infrastructure for communication and their positions are not pre-determined.

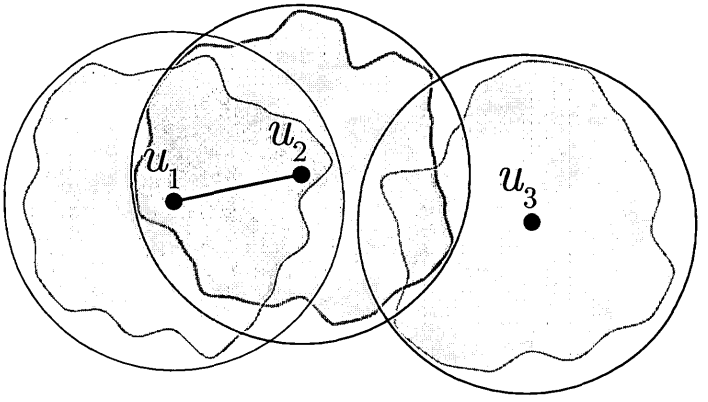


Figure 1.1: A set of three wireless devices with irregular communication regions. The circle of radius r about the position of each node represents the limit on the communication range. Node u_1 and u_2 are connected and can exchange messages because each is positioned within the other's communication region. Node u_3 is not connected to the other two nodes.

The approach adopted for positioning will be influenced by the limitations of a sensor network. A few criteria that will impact a positioning mechanism are

identified below. These will help provide some scope for modelling a mobile wireless communication system under which the positioning mechanisms of interest must operate.

Constraints: A positional constraint is a condition on the Euclidean distance between the position of two nodes. There are many ways to constrain the positions of nodes. This work limits the constraints to be imposed between two nodes where a signal is wirelessly transmitted from one node and received at another node. These are the basis constraints and they can be used to derive other constraints. The reason for this limitation is that constraints can be imposed on the distance between nodes based on properties of wireless communication, a basic capability of every node. Studying the positioning problem with only basic hardware tools provides insight into the performance limits of a positioning mechanism.

Computation: The aspect that differentiates sensor devices from more general wireless devices is that sensors operate with extreme resource limitations. The computational limitations are not precisely defined but they make the positioning problem challenging and render some positioning methods impractical. There are two extremes in which this computation can occur: centrally at a single node or distributed across all nodes. Given the limited computational resources of sensor devices, it is tempting to use solutions that involve centralized computation. However one requirement of the positioning mechanism for a sensor network is that the

constraint information between pairs of nodes is locally determined, after the algorithm has started. That is to say, the acquisition of constraints is a local process and is determined by the exchange of messages between nodes. Centralized algorithms that collect all constraint information at a single node to compute the node positions are admissible in this model, but they suffer from scalability issues. The amount of distributed computation by a positioning algorithm is especially important when nodes are mobile since the communication links and node positions can change significantly, depending on the node speed, before the algorithm terminates.

Seed Nodes: A seed node (sometimes referred to as an anchor node) is one that always knows its true position. The existence of seed nodes helps a positioning algorithm considerably. A seed node may be implemented in many possible ways. A set of stationary nodes could be used as seed nodes if their position is known. Alternatively some nodes may be equipped with a Global Positioning System (GPS) unit that periodically determines the position of the node.

Mobility: A mobile node is capable of changing its position over time. This implies that an estimate of a node's position is regularly out of date. In addition to the dated position estimates, node mobility complicates many positioning algorithms because the pairs of nodes that can communicate changes over time. This sensor network property is perhaps the most significant because a computationally intensive algorithm may quickly exhaust the limited resources of a device. When

the nodes are stationary it may be argued that the node positions only need to be estimated once and therefore a computationally intensive algorithm which achieves good quality position estimates is justified. In a mobile scenario, the expensive algorithm may become infeasible.

The determination of geographic position has been studied for many years and there may be an existing solution that is suitable for mobile sensor networks. Section 1.3 discusses alternative solutions to the positioning problem for sensor networks that invoke the use of a positioning mechanism that is external to the network. Section 1.4 presents the Network Positioning Problem which is better suited to sensor networks. This is an approach to positioning with greater autonomy than an auxiliary positioning method like the Global Positioning System.

1.3 Auxiliary Positioning Methods

An auxiliary positioning mechanism is a means of determining the positions of sensor nodes via the use of a resource that is external to a sensor network. The positioning of sensor nodes is then reliant on this external resource.

One possible solution to the positioning problem is the manual assignment of positions to nodes as they are deployed. The determination of the position of each sensor node can be made by the Global Positioning System or surveying techniques, for example. This solution can be afforded in some cases where a network is known

to be stationary. The manual placement of nodes can be time consuming if the network is large, which is expected in many scenarios. When nodes can change their position or the network is deployed in a way that makes the resting position of a node unknown, as may be the case with many potential applications of sensor networks using randomized deployments, this option cannot be adopted.

Another possibility is to equip each node with a Global Positioning System (GPS) receiver. Originally designed for military use, the Global Positioning System, partially illustrated in Figure 1.2, is a system of satellites that orbit the earth and continually transmit messages that can be received by a GPS receiver. A GPS receiver can determine the position and distance to the satellite that sends a message based on the message's time of flight. With communication from four different satellites the receiver can determine its position by an operation known as **multilateration**.

The use of GPS receivers in practice is becoming extensive with the growing use of navigational aides in mobile phones and automobiles. Despite the growing use of GPS in consumer electronics, there are many disadvantages to the system. One drawback of GPS is that the positioning services of a sensor network are dependent on an external system. The GPS receiver also requires a line-of-sight to many satellites for proper operation making their use for indoor environments infeasible. Most importantly however, the sensor devices introduced in the beginning of this

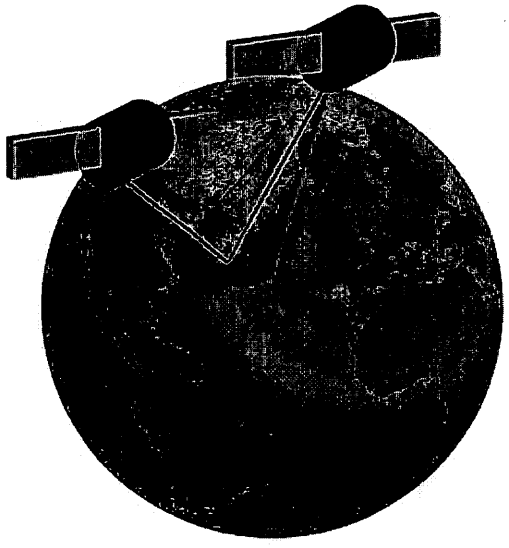


Figure 1.2: Global Positioning System: The distance from two satellites to a point on the surface of the earth. A GPS receiver requires communication from at least four satellites to resolve its own position: three of the satellites are used for positioning the receiver and a fourth satellite is used for timing.

chapter are physically small and have extremely limited energy resources. GPS receivers have been found to require considerable amounts of energy relative to other components of the sensor device, as revealed by the LynxNet sensor network example. Since a GPS receiver is considered to be energy intensive, it is a poor choice as the entire basis of a positioning solution for such resource limited devices. In Section 1.4 a positioning mechanism is described that is better suited to sensor networks.

1.4 The Network Positioning Problem

A positioning mechanism that makes use of the intrinsic properties of a sensor network is solving the network positioning problem where the structure of the network is used as a basis for positioning. Section 1.4.1 describes the basic model under which a network positioning algorithm will operate. Although the basic model admits many possible algorithms, it will become clear that some of the algorithms are inadequate for positioning in the case of sensor networks.

1.4.1 Formulation of the Positioning Problem

The environment that models the possible node positions is two-dimensional Euclidean space $\mathbb{R}^2$ and the distance between two points $x, y \in \mathbb{R}^2$ is the Euclidean distance, denoted $d(x, y)$. Throughout this work we use the notation $|\cdot|$ to denote

the number of elements in a set. If V is the set of nodes in the system, then the function

$$p : V \to \mathbb{R}^2, \tag{1.1}$$

called the **embedding**, gives the true position of each node u_i in $\mathbb{R}^2$. If $p(u_i)$ is the position of node u_i, then $x_p(u_i)$ denotes the x-coordinate of its assigned position and $y_p(u_i)$ denotes the y-coordinate of its assigned position.

The **communication region** of a node is the set of positions where a transmitted message by one node can be received by another node. Figure 1.1 illustrates the concept of this region of space. The elements of the set $N(u_i, 1)$ are the nodes from which a message is wirelessly received at node u_i and are called **neighbours**, or first neighbours, of node u_i. The **degree** of a node is the number of first neighbours $|N(u, 1)|$. The elements of the set $N(u_i, 2)$ are the nodes from which a message is wireless received at a neighbour in $N(u_i, 1)$ but not received at node u_i and are referred to as **second neighbours**.

The neighbours of each node induces a directed **connectivity graph** (V, E), where V is the set of nodes and the edge $(u_i, u_j) \in E$ if $u_i \in N(u_j, 1)$. Unless otherwise stated, the connectivity graph is undirected so that $(u_i, u_j) \in E \leftrightarrow (u_j, u_i) \in E$. A **path** is a sequence of distinct nodes $u_1, \ldots, u_k$ such that $\{(u_1, u_2), (u_2, u_3) \ldots, (u_{k-1}, u_k)\} \subseteq$

E and the length of the path is the number of edges of the path.

A **line segment** between two points x and y in $\mathbb{R}^2$ is the set of points $\mathcal{L} = \{(1-\alpha)x + \alpha y : \alpha \in [0,1]\}$. The **circle** of radius r about point x in $\mathbb{R}^2$ is the set of points $\mathcal{C}(x,r) = \{z : d(x,z) = r\}$. A (closed) **disk** of radius r about point x in $\mathbb{R}^2$ is the set of points $\mathcal{D}(x,r) = \{z : d(x,z) \leq r\}$.

An asymmetric **lens** $\mathcal{I}(r_1, r_2, s)$ is a region formed by the intersection of two disks $\mathcal{D}(x,r_1)$ and $\mathcal{D}(y,r_2)$ with radii r_1 and r_2, respectively, whose centres are separated by a distance s. The shaded region in Figure 1.3 is an asymmetric lens. When $r_1 = r_2$ the lens is symmetric.

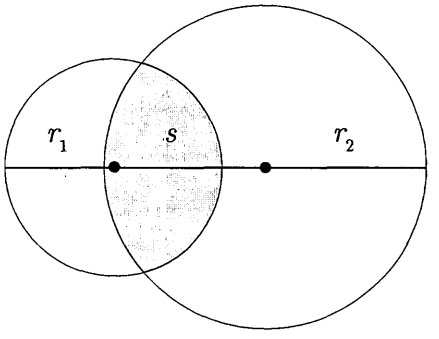


Figure 1.3: An asymmetric lens.

A **constraint** is a condition on the Euclidean distance between a pair of nodes. The network positioning problem is to find an assignment $q : V \to \mathbb{R}^2$ subject to the constraints given by each pair of nodes in the set E. A positioning algorithm takes the constraints on node positions as input and computes a position assignment q. If

the position assignment q satisfies all the constraints it is called a **realization**. The **positional error** of a node is $d(p(u_i), q(u_i))$, denoted $e_q(u_i)$, and is a performance measure of a positioning algorithm. This performance measure is widely used in the literature [67, 63, 12].

Perhaps the simplest type of constraint between a pair of connected nodes is derived from the communication range r. The constraint is imposed on a pair of nodes based on the assumption that the nodes can communicate only if their distance is less than or equal to r. This is referred to as a connectivity-based constraint.

A more sophisticated technique would try to estimate the distance between the pair of nodes. Many methods exist for the distance measurement, although they either use hardware or additional processing beyond what is required by connectivity-based constraints. For example, if nodes have synchronized clocks, and know the travelling speed of the signals used for communication, then pairs of communicating nodes can estimate the distance between themselves using a distance-speed-time expression. Some authors have modelled these inter-node distances exactly and have shown that for some combinations of constraints, the node positions are uniquely determined. Section 1.4.2 discusses some of this work in greater detail. Unfortunately distance measurement techniques involve measurement error, so the exact distance determination is not possible in reality. This has led other researchers

to consider the more relaxed connectivity-based constraints that take the form of upper bounds on inter-node distances.

These are two basic forms of distance information between pairs of nodes that will constrain their positions: distance equalities, which are discussed in Section 1.4.2, and distance inequalities which are discussed in Section 1.4.3.

1.4.2 Constraints as Equalities

A **framework** (V, E, p) is a graph (V, E) of n stationary nodes $\{u_1, u_2, ..., u_n\}$, the edges E represent distance constraints between pairs of nodes, and p is the embedding of the nodes. Each edge $(u_i, u_j) \in E$ is associated with the distance $d(p(u_i), p(u_j))$, which is a constraint on the distance between nodes u_i and u_j. Two frameworks (V, E, p) and (V, E, q) are **equivalent** if $\forall (u_i, u_j) \in E$, $d(p(u_i), p(u_j)) = d(q(u_i), q(u_j))$. Two equivalent frameworks are **congruent** if $\forall u_i, u_j \in V$, $d(p(u_i), p(u_j)) = d(q(u_i), q(u_j))$. If a graph G is such that all embeddings are congruent to one another, under translation, rotation and reflection, then the graph is said to have a **unique realization** in the plane because no matter which way the graph is embedded the distance between any pair of points is the same.

An embedding is **generic** if the node positions are algebraically independent. Almost all embeddings are generic [19]. A framework (V, E, p) is **generically rigid** if there exists a $\epsilon > 0$ such that if (V, E, p') is equivalent to (V, E, p) and

$d(p(u_i, p'(u_i)) < \epsilon \; \forall u_i \in V$, then (V, E, p') is congruent to (V, E, p) [25]. A framework is **globally rigid** is every realization of (V, E) is a unique realization.

A generically rigid framework can have many equivalent realizations that are not congruent. Two different types of graph configurations, known as **binary ambiguities**, result in equivalent realizations that are not congruent and are illustrated in Figure 1.4.

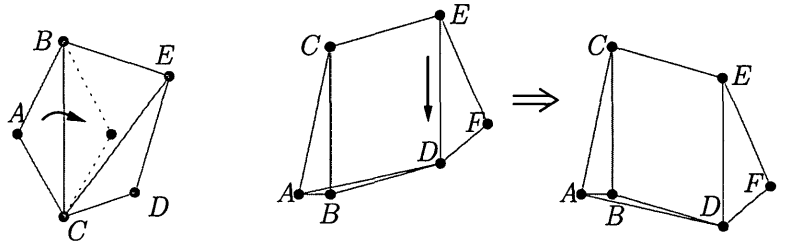


Figure 1.4: Binary Ambiguities in Generically Rigid Frameworks: A flip ambiguity (left), where node A has two possible positions, and a discontinuous flex ambiguity (right), where the removal of edge (A, D) creates a flexible graph allowing the node positions of D, E, F to be deformed into an alternative configuration where edge (A, D) can be reinserted [42]. Notice these ambiguities result in equivalent realizations since the constraints are satisfied, but the realizations are not congruent since the distances between some non-adjacent pairs of nodes have changed.

A graph (V, E) is **redundantly rigid** if, after removing an edge from E, the framework (V, E, p) remains generically rigid. A graph (V, E) is 3-connected if it remains connected after removing any two vertices. A framework is generically globally rigid if and only if either (V, E) is a complete graph on at most three

vertices, or the graph (V, E) is redundantly rigid and 3-connected [25]. In other words, when the embedding is generic, the uniqueness of a framework (V, E, p) is determined by properties of the graph (V, E). When a framework is generically globally rigid, the graph (V, E) is referred to as a generically globally rigid graph.

A generically globally rigid graph has a unique realization, up to translations, rotations and reflections, but finding such a realization, given the graph, has been shown to be NP-Hard [2]. The realization problem remains NP-Hard even for generically globally rigid graphs where $(u_i, u_j) \in E \leftrightarrow d(p(u_i), p(u_j)) \leq r$, graphs that more closely match the connectivity graphs of the network localization problem. A sub-class of generically globally rigid graphs called **trilateration graphs** are efficiently solvable. A trilateration graph is characterized by the trilateration graph operation in two-dimensions: given a generically globally rigid graph with three or more nodes the addition of a node and three edges to unique nodes preserves global rigidity. A graph obtained by a sequence of trilateration operations is a trilateration graph [2]. A node can be positioned using a trilateration, as illustrated in Figure 1.5, and an entire graph can be position by performing a sequence of trilateration operations.

Unfortunately, accurately measuring inter-node distances is difficult in practice as it involves measurement error which leads to robustness issues with the constraints. The next section relaxes the requirement of exact distances for con-

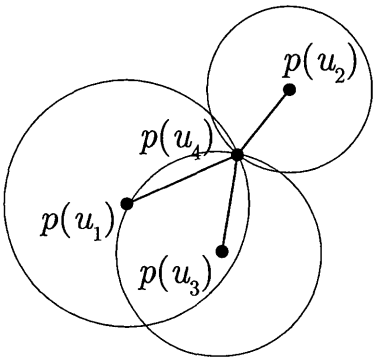


Figure 1.5: Trilateration operation: Calculating the exact position of a node u_4 in two dimensional space requires three distances to three other nodes, here u_1, u_2, and u_3, of known positions $p(u_1)$, $p(u_2)$, and $p(u_3)$, respectively.

straints.

1.4.3 Constraints as Inequalities

Suppose $(u_i, u_j) \in E \rightarrow d(p(u_i), p(u_j)) \leq r$. The restriction that $d(q(u_i), q(u_j)) = d(p(u_i), p(u_j))$ for each pair of nodes $(u_i, u_j) \in E$ can be relaxed to $d(q(u_i), q(u_j)) \leq r$ if $(u_i, u_j) \in E$. This information can be used to impose an inequality constraint on the distance between a pair of nodes.

The communication region of each node defines the connected nodes in the network and provides a natural mechanism for defining the constraints: so long as a node knows its communication range r, the reception of a transmitted message is enough to define a constraint on the distance between the two nodes. These

are **connectivity-based constraints**. The fixed value r is an idealization of the transmission range of a radio signal. Algorithms often assume that $(u_i, u_j) \notin E \rightarrow d(p(u_i), p(u_j)) > r$ to impose further constraints on node positions.

A commonly studied model for wireless communication is the Unit Disk model where $(u_i, u_j) \in E \leftrightarrow d(p(u_i), p(u_j)) \leq r$. In this model the communication region of each node is a perfectly shaped disk and the connectivity graph induced by the disks is called a **unit disk graph**. This model provides a useful framework for conceptualizing wireless networks and is accessible to analytic techniques.

There has been some theoretical work on the difficulties of the positioning problem for the unit disk model of communication. It has been shown that even in this relaxed version of the problem, determining if a graph is a unit disk graph is NP-hard [8].

1.4.4 Information for Connectivity-based Positioning Algorithms

This dissertation investigates connectivity-based positioning algorithms. This class of positioning algorithm uses node connectivity to impose constraints on node positions. The algorithms under consideration are executed on each node and the node connectivity is learned through the exchange of messages between nodes. The communication range r and the maximum movement distance per time step δ are known to each node.

It is common to assume the existence of a number of special nodes called **seed nodes** that always know their true position in the space in which they are embedded [30, 21, 67]. The seed nodes may acquire their position via manual placement, GPS, or some other means.

1.5 Overview of Dissertation

The work in this dissertation investigates both theoretical and applied aspects of the connectivity-based positioning problem for mobile sensor networks. The models under which the problem is studied are described in Chapter 2. The nature of node deployment and node movement result in an embedding of nodes that is uncertain. Likewise, random mechanisms are involved in the communication process between nodes. The uncertainty involved in these processes is modelled with techniques from probability theory.

There has been extensive theoretical and applied work on the connectivity-based positioning problem. The current research on connectivity-based positioning for sensor networks is described in Chapter 3.

The set of feasible position assignments for a node is a recurring object in this work since it is related to positional error. Chapter 4 formulates an expression for the positional error that is inherent to the models of the sensor network. The analysis uses a standard communication and mobility model. The formulation is

extended to an analysis of the effectiveness of accumulated constraints on node positions due to node mobility.

In Chapter 5 a new probabilistic communication model is developed based on observed communication properties of sensor devices. This model is used later for experimentation.

A new positioning algorithm for mobile sensor networks is proposed in Chapter 6. The algorithm exploits a property of disk graphs to impose new constraints on node positions. The new constraints reduce the size of the set of feasible positions for a node. The algorithm also characterizes the shape of the set of feasible positions. Some feasible sets are composed of isolated components which can lead to the propagation of large positional errors. The Orbit algorithm uses a heuristic technique to address this problem. The performance of the Orbit algorithm is evaluated using computer simulations and compared against another positioning algorithm. Chapter 7 describes the implementation of the Orbit algorithm on resource limited hardware and the performance of Orbit on deployments of sensor networks.

Chapter 2

Modelling Mobile Wireless Networks

The work in this dissertation considers a system of computing devices called sensors that are capable of wireless communication. Each sensor is positioned in an environment and nearby devices can communicate by the exchange of messages. The devices are mobile so the position of a sensor can change over time. The models for node position distribution, communication and movement are given below.

2.0.1 Node Distribution in a Space

The sensor devices are represented by the set of nodes V and the physical environment in which the sensors are located is modelled as a two-dimensional Euclidean space $\mathbb{R}^2$. The space is an abstraction of an environment that is a flat surface. A bounded region of $\mathbb{R}^2$ is used when the environment has boundaries. It is assumed

that there is an embedding $p_t : V \to \mathbb{R}^2$ that gives the position of each node at a point in time t, and the distance $d(p_t(u), p_t(v))$ between a pair of nodes u and v is the Euclidean distance.

Many node deployment scenarios involve a random mechanism that leaves the positions of the nodes in an uncertain state. For example it has been proposed that some sensor networks be deployed into the environment by dropping from the aircraft [20]. This dissertation models the embedding of the sensors as a random process and focuses on properties that arise from typical random embeddings since these are interest to designers of sensor networks. The use of random node distributions is commonplace in the literature on wireless ad hoc networks.

A homogeneous Poisson point process on $\mathbb{R}^2$ is a counting process used to model the variability in the number and position of the nodes. This distribution imposes no structure on an embedding, which is common when analyzing positioning algorithms [35, 2, 59, 61]. The complete spatial randomness ensures all configurations of n nodes embedded in the space are equally likely [17]. This characteristic is important for unbiased coverage of an environment but it can also serve as a reference model for more complicated scenarios that involve non-homogeneous configurations of nodes.

An embedding is a set of nodes with positions, and is characterized by the random quantity $X_t(\mathcal{X})$, which is the number of nodes in a bounded measurable

set $\mathcal{X} \subset \mathbb{R}^2$ at time t. If X_t is a Poisson process with intensity λ and $\mu(\mathcal{X})$ be the area of $\mathcal{X}$, the probability of k nodes being contained in $\mathcal{X}$ is

$$\mathbb{P}[X_t(\mathcal{X}) = k] = \frac{e^{-\lambda \cdot \mu(\mathcal{X})}(\lambda \cdot \mu(\mathcal{X}))^k}{k!}. \tag{2.1}$$

The Poisson formula shows that bounded sets of equal size in $\mathbb{R}^2$ have an equal probability of being a location for a number of nodes.

2.0.2 Node Communication

The **communication region** of a node is the set of positions where a transmitted message by one node can be received by another node. The model of communication determines whether two nodes are connected at time t. The elements of the set $N(u_i, 1)$ are the nodes from which a message is wirelessly received at node u and are called **neighbours**, or first neighbours, of node u_i. The elements of the set $N(u_i, 2)$ are the nodes from which a message is wireless received at a neighbour in $N(u_i, 1)$ but not received at node u_i and are referred to as **second neighbours**. The neighbours of each node at a time t induces a directed **connectivity graph** where the nodes V are the graph vertices and the graph edges are the set of node pairs $\{(u_i, u_j) | u_i \in N(u_j, 1)\}$.

This dissertation considers three models of communication: Unit Disk (UD),

Noisy Disk (ND) and Empirical Disk (ED). Each model captures the approximately disk shaped region of communication of a radio with an isotropic antenna where a signal is emitted equally in all directions.

The **Unit Disk** model defines the communication region of a node u_i by a disk of radius r about its position $p(u_i)$. Any node u_j that is positioned within the disk is connected to node u_i. The connectivity graph resulting from this model is known as a **unit disk graph** and has been used extensively in modelling wireless network communication [61, 21, 67].

The **Noisy Disk** model generates communication regions that are more general than the perfectly disk-shaped communication region. It uses an irregularity parameter $\alpha \in [0, 1]$ that determines a random communication range from the interval $[(1 - \alpha) \cdot r, (1 + \alpha) \cdot r]$ for each direction [21].

The **Empirical Disk** communication model generates the communication properties from real sensor devices. Two different sensor devices were used over the course of this work and an empirical disk model was generated for each device. The communication model based on Texas Instruments hardware is denoted $\mathrm{ED_{TI}}$ and the communication model base on Nordic hardware is denoted $\mathrm{ED_N}$. These probabilistic models are based on observed connectivity from sensor devices and their derivations are described in Section 5.1.

2.0.3 Node Movement

The positions of the nodes change over time. Node movement is regulated by a discrete time system. In each time step a node communicates with its neighbours, performs some computation, and moves to a new position that is at most a distance δ from its previous position. This system induces a sequence of connectivity graphs $\{(V, E_t) : t \geq 0\}$ where the edges E_t are induced by the node connectivity at time t. Two models are considered for node movement: Random Waypoint (RP) and Random Walk (RW). In each model the movement of a node independent of any other node.

In the **Random Waypoint** model each node selects a destination point from the space $\mathbb{R}^2$ and moves along a line towards it. In each time step the node moves a random distance chosen uniformly from the interval $[0, \delta]$ on this line. The travel distance is truncated if it would exceed the destination point. When the node reaches the destination it selects a new destination point for the next time step. The node pauses for a duration of time between arriving at a destination and departing for its new destination. This dissertation considers the case where the pause duration is zero, which is the most studied version of the model.

In the **Random Walk** model each node moves a uniformly random distance selected from the interval $[0, \delta]$ in a random direction in each time step. When the

step lengths fixed in length, the random walk is also known as Pearson's Walk.

Both movement models have been utilized extensively in previous work and have been implemented in computer simulations [10, 45, 65, 43]. Though each process for each model seems similar since the movements of each node are independent and each node moves in a direction for a period of time, the dynamics of each model are very different. Some of the differences are discussed in Chapter 6.

Chapter 3

Literature Related to Connectivity-based Positioning

There has been a wide variety of algorithms proposed for finding an assignment of node positions when they are constrained by the structure of the disk graph. The subject of positioning for sensor networks has been surveyed many times in the literature [11, 31, 1]. Due to the hardness of the positioning problem, algorithms proposed for positioning do not in general find solutions that satisfy all the constraints. A performance measure used by many algorithm designers is the positional error of the position assignment of a typical node. The positional error of a node u_i is defined to be $d(p_i, q_i)$, the distance between p_i, the true position of a node, and q_i, the position assigned by an algorithm.

Many algorithms are based on the shortest graph paths between a pairs of nodes. The concept is illustrated beginning with the most basic algorithms in Section 3.2. Some shortcomings of this approach are identified and traced back to the graph structure. The challenges posed by using graph paths alone are addressed by using more sophisticated graph structures. Section 3.3 describes a technique from statistics has been used to address the positioning problem. The standard path-based approaches are not suitable for systems of mobile nodes because the graph structure is continually changing which makes path determination difficult. The difficulties are accounted for by positioning algorithms designed for the case of mobile nodes which are described in Section 3.4. These algorithms use more local graph information for positioning and take advantage of an opportunity presented by mobility: historical constraint information can be used to reduce the error of a node's current position. Comparatively less research effort has been expended deriving bounds on positional error than on algorithm development. Section 3.5 concludes this chapter with a presentation of some analysis of some factors that affect the accuracy of connectivity-based positioning algorithms.

While this work examines positioning algorithms that operate with connectivity imposed constraint information, it should be noted that some of the proposed algorithms can operate under a stronger model where a node can estimate the distance between itself and its neighbour nodes.

3.1 Overview of Algorithms

The algorithms under consideration in this chapter operate with the connectivity-based constraint information defined by the connectivity graph (V, E), as described in Section 1.4.3. Another common assumption is that m of the n nodes are seed nodes. This allows the algorithm to assign positions that are consistent with the global coordinate system - information that is stored by the seed node positions. Many algorithms also assume each node is assigned a unique label, called an identifier.

The algorithms covered in this survey are assumed to be operating in discrete time. In each time step t each node u_i:

1. Exchanges messages with its neighbours $N(u_i, 1)$ at time t,
2. Computes a position estimate $q_t(u_i)$, and
3. Moves to a new position that is at most a distance δ from its current position.

Note that, if the nodes are stationary (i.e., they do not change their position in each step), then Step 3 is omitted.

What makes a good connectivity-based positioning algorithm? One must consider the platform on which an algorithm is executed. While the tolerance for positional error will vary depending on the particular application, the computational

complexity and the communication complexity may be far more critical design parameters for devices with limited resources.

The positioning problem can be formulated as a non-convex optimization problem and approached using global optimization techniques like simulated annealing. There are, however, two drawbacks to these approaches: (i) they are computationally expensive, and (ii) they do not yield much insight into the connectivity-based positioning problem. If the optimization problem is restricted to consider only convex constraints then the problem can be formulated as a convex optimization problem and efficiently solved using Semidefinite Programming (SDP) [6]. This technique is considered computationally intensive however since its worst case running time is $O(n^6)$, for a sensor network of n nodes [5]. When the node density increases, the number of constraints does as well so the complexity can decrease to $O(n^3)$. When running the SDP-based algorithm on a modern desktop computer, the authors state:

> "While we could solve localization problems with 50 sensors in few seconds, we have tried to use several off-the-shell [sic] codes to solve localization problems with 200 sensors and often these codes quit either due to memory shortage or having reached the maximum computation time" [5].

The addition of node mobility compounds the problem of expensive algorithms since the node positions are changing frequently. Biswas and Ye have proposed a modified SDP-based algorithm that is performed on clusters of nodes to mitigate

this problem of exceptional memory requirements [6]. Still, given the limited resources available, less computationally expensive algorithms are needed for practical approaches to positioning in the domain of sensor networks.

3.2 Path-based Algorithms

Perhaps one of the most trivial positioning algorithms proposed is the Centroid method [9]. In every time unit each seed node sends its position to its neighbour nodes. Each non-seed node u_i that receives messages from $0 < m' \leq m$ seed nodes computes its position

$$q(u_i) = \left(\frac{\sum_{j=0}^{m'} x_p(u_j)}{m'}, \frac{\sum_{j=0}^{m'} y_p(u_j)}{m'} \right),$$

where the point $(x_p(u_j), y_p(u_j))$ is the position received from seed node u_j. Schemes like Centroid are known as one-hop algorithms since only neighbour information is used. Such a design prevents positional information originating at seed nodes from propagating throughout the network. Consequently, in our defined model, one-hop algorithms can require high seed node densities to achieve acceptable node position estimates. There is more positional information within the graph beyond neighbour nodes, as the next section reveals.

3.2.1 DV-Hop

Initially proposed by Niculescu and Nath, the DV-Hop algorithm generalizes the idea of estimating the distance to seed nodes that was proposed in the Centroid algorithm by using graph paths to estimate the Euclidean distance to seed nodes [46, 47]. This algorithm is one of the most widely cited hop-based positioning algorithms and many papers that propose new positioning algorithms, sometimes even based on DV-Hop, still compare performance with the original version of the algorithm [68, 34, 30].

The original idea behind the algorithm was to adopt the mechanism of the GPS: obtain distance measurements to at least three nodes of known position and use these distances to find a position assignment for a node of unknown position [46].

DV-Hop Algorithm Outline:

1. Each node learns the position q_i and shortest path length h_i to each seed node u_i by message passing

2. Each seed node u_i computes the average distance per hop,

$$c_i : \frac{\sum d(q(u_i), q(u_j))}{\sum h_i}, \forall \text{ seed nodes } u_j \neq u_i$$

3. Each node receives the mean distance per hop c_i from its nearest seed node

u_i.

4. Each node u_i performs a least squares computation to find its position $q(u_i)$ using the estimated distance $\hat{d}_{i,j} = c_i \cdot r$ to seed node u_j.

Rather than use the upper bound on Euclidean distance to each seed node, which is r times the shortest path length to the seed node, DV-Hop tries to estimate the distance $\hat{d}_{i,j}$ from node i to seed node j using the mean distance per hop c_i.

The final steps of the algorithm estimates Euclidean distance to the seed nodes and finds a position assignment q_j for node u_j that minimizes the squared error,

$$\sum_{i=1}^{m} (d(q(u_i), q(u_j)) - \hat{d}_{i,j})^2.$$

The set of equations can be solved for $q(u_i) = (x_q(u_i), y_q(u_i))$ using a least squares method as described by Langendoen and Reijers [29]. A sequence of algebraic operations can be performed on the the system of equations making them linear in $x_q(u_i)$ and $y_q(u_i)$,

$$A = \begin{bmatrix} 2(x_q(u_1) - x_q(u_m)) & 2(y_q(u_1) - y_q(u_m)) \\ \vdots & \vdots \\ 2(x_q(u_{m-1}) - x_q(u_m)) & 2(y_q(u_{m-1}) - y_q(u_m)) \end{bmatrix}$$

$$b = \begin{bmatrix} x_q(u_1)^2 - x_q(u_m)^2 + y_q(u_1)^2 - y_q(u_m)^2 + d_m^2 - d_1^2 \\ \vdots \\ x_q(u_{m-1})^2 - x_q(u_m)^2 + y_q(u_{m-1})^2 - y_q(u_m)^2 + d_m^2 - d_{m-1}^2 \end{bmatrix}.$$

The system is solved for $q(u_i) = (x_q(u_i), y_q(u_i)) = (A^T A)^{-1} A^T b$, provided the inverse of matrix A exists [29]. This algebraic technique is simpler than the proposed gradient decent methods for finding the optimal solution to this system of equations [46, 3, 44].

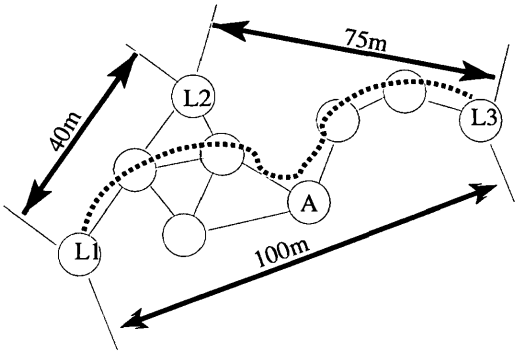


Figure 3.1: DV-Hop uses shortest graph paths to estimate the distance between pairs of nodes [46].

Another positioning algorithm very similar to DV-Hop is Amorphous, which was initially proposed by Nagpal et al. [44]. The first step of this algorithm is the same as DV-Hop: each node finds the shortest path to at least three seed nodes. In contrast to the DV-Hop algorithm, Amorphous computes the estimated distances

to seed nodes using the node density λ.

After a node has shortest path length h_i to each seed node u_i, it exchanges h_i with its neighbours and computes the average

$$\overline{h}_i = \frac{\sum_{j \in N(u_i)} h_j + h_i}{|N(u_i)| + 1} - \frac{1}{2}$$

as the mean path distance to seed node u_i. The network wide average distance per hop is

$$c = r \cdot (1 + e^{-\lambda \pi r^2} - \int_{-1}^{1} e^{-\lambda r^2 (\arccos t - t\sqrt{1-t^2})} dt),$$

which is the Kleinrock-Silvester formula for the expected distance per hop when the node positions are Poisson distributed with intensity λ [28]. This technique saves many rounds of message passing by avoiding the second broadcast phase of the DV-Hop algorithm to distribute the calculation of the average distance per hop c_i at each seed node u_i.

Inspection of the Kleinrock-Silvester formula reveals a conclusion that one might intuitively expect: the mean distance per hop increases as the node density λ increases and approaches one, as revealed by Figure 3.2. By definition, the maximum distance between a node and any of its neighbours is r. Increasing the node density increases the probability of a more distant neighbour node. This is significant

since the higher the value of c, the closer the estimated distance $\hat{d}_{i,j}$ is to the true Euclidean distance between nodes u_i and u_j. Based on this formula, the authors suggest the expected number of neighbours, $\lambda\pi r^2$, should be 15 to achieve good estimates of the distance to seed nodes and therefore reasonable positional accuracy. The authors go on to suggest that, based on observations from computer simulations, 10 is the minimum expected number of neighbours needed to have a reasonable chance of avoiding isolated nodes. Note that the Amorphous algorithm assumes the node density λ is known prior to the execution of the algorithm.

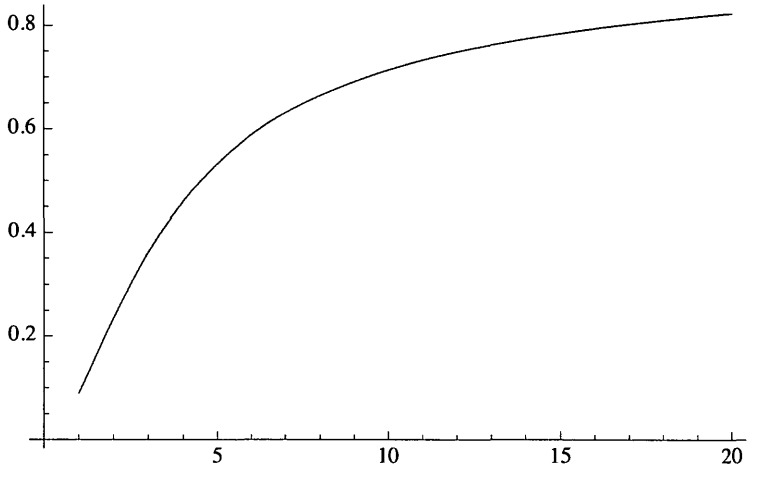


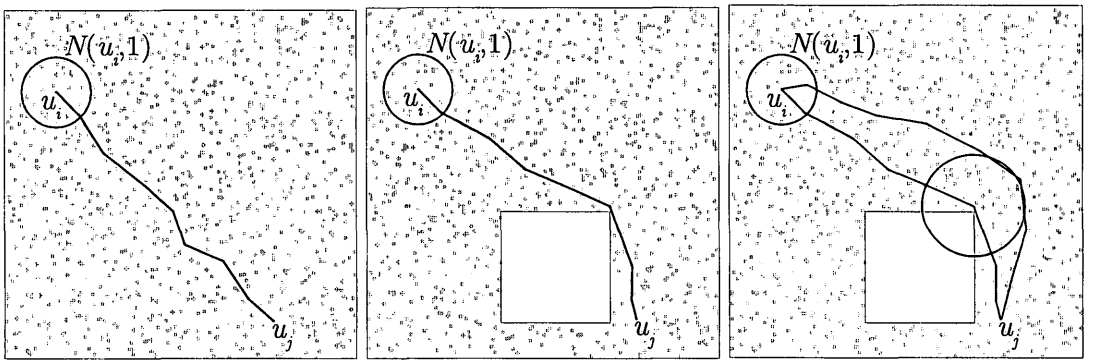
Figure 3.2: Kleinrock-Silvester Formula: The Mean Euclidean Distance per Hop. Values on the x-axis are the expected sizes of the node neighbourhoods.

3.2.2 Sparse and Irregular Node Embeddings

The Kleinrock-Silvester formula shows the mean distance per hop increases with node density. When the node density is low, the mean distance per hop is much smaller than the radio range r and the graph distance increases. Therefore, if the node density is too low, the graph distance can be a poor estimate of the Euclidean distance. Irregular point configurations as illustrated in Figure 3.3, perhaps due to obstacles in the environment, also increase the mean positional error of these algorithms since the shortest path lengths further obscure the true distance between nodes.

Both situations result in curved, as opposed to straight, paths to seed nodes. This issue as it pertains to low node density is discussed in further detail in Section 3.2.3. When the curvature is due to an irregular embedding, the accuracy of algorithms can be improved by detecting the boundaries of the network and measuring the distance between nodes using the composition of several straight-line paths.

One of the central criticisms of the DV-Hop approach to positioning is its failure to operate well when the embedding of the nodes is irregular. This irregularity could be due to obstacles within the environment which create regions where nodes are prohibited from residing, or simply due to a non-uniform deployment. In either case



(a) (b) (c)

Figure 3.3: A hole in the node embedding that affects the path-based Euclidean distance estimation of DV-Hop. Figure 3.3a illustrates a hole-free embedding where the shortest path length is a good estimate of the Euclidean distance between nodes u_i and u_j. Figure 3.3b presents an with hole that forces the shortest path to bend which results in a less accurate estimate of the distance between the nodes. Figure 3.3c represents the detection of the nodes on the boundary of the hole and decomposing the shortest path between the nodes into several paths [30].

the result is that the shortest path length is a poor estimate of the straight-line distance between a pair of nodes since the path bends around irregularities within the node configuration. Li and Liu address this problem by identifying nodes in close proximity to environmental boundaries [30]. If the boundary nodes appear on the shortest path between a pair of nodes, then the path is partitioned into multiple paths terminated at the boundary nodes. The estimated distance between a pair of nodes becomes a calculation involving many straight-line distances to account for the indirect nature of the shortest path.

3.2.3 Improving Distance Estimates

All of the previously described algorithms share a commonality: they use simple graph paths as the sole combinatorial structure to impose constraints on node positions. There are other structures available that can assist with the layout of the nodes as O'Dell et al. illustrate with their GHoST algorithm [48].

As previously stated, one of the limitations of graph paths for estimation of the Euclidean distance between a pair of nodes is that the nodes composing a path never fall on a straight line. The curvature can be due to environmental obstacles requiring paths to bend around holes in the network. In this case, holes can be handled by a positioning algorithm that detects the boundaries of a network. Another anomaly that affects the straightness of graph paths is low node density.

Since the path length is only an upper bound on Euclidean distance it can be difficult, even with averaging, to get good estimates of the distance between a pair of nodes. It is this latter issue that is addressed by the authors of GHoST by making better use of the available constraint information.

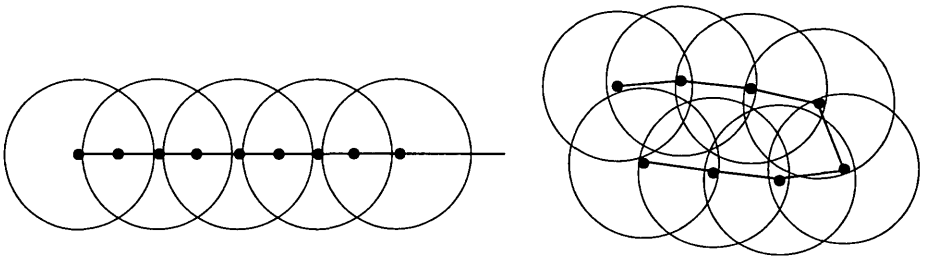


Figure 3.4: Bending: The difficulty with using individual path structures in two-dimensional Euclidean space [48].

A new structure known as a trimmer is introduced in GHoST which integrates graph paths into new objects that yield an improved upper bound on the Euclidean distance between a pair of nodes [48].

A trimmer is based on the following observation: if u, v, w and u, x, w are the shortest paths between nodes u and w, and edge $(v, x) \notin E$, then $d(p(u), p(w)) \leq \sqrt{3}$. This is in contrast to the constraint $d(p(u), p(w)) \leq 2$ when just one of the aforementioned paths is considered. This can be generalized to the shortest paths $uv_0 \ldots v_k w$ and $ux_0 \ldots x_k w$ between node u and node w where $(v_0, x_0), (v_k, x_k) \notin E$ and $(v_i, x_j) \notin E$ for $i \neq j$. When these paths exist $d(p(u), p(v)) \leq k + \sqrt{3}$.

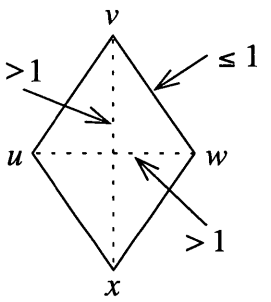


Figure 3.5: The basic structure for the trimmer that reduces the distance between the nodes from 2 to $\sqrt{3}$ by considering multiple paths between node u and w [48].

The authors suggest that existing hop-based algorithms be augmented with these structures to improve their performance.

3.3 Multidimensional Scaling

Multidimensional Scaling (MDS) is a set of statistical techniques often used as a means of visualizing data. The process takes proximity information between data points and maps them to lower dimensional spaces to assist in data analysis. In some sense this is precisely what the positioning problem requires so it is no surprise that this technique has been proposed as a solution to the positioning problem. The approach has the nodes as data points and the proximity information is derived from shortest path lengths between pairs of nodes. The MDS technique is applied to these inputs to produce a map of the nodes to a two dimensional space. The

solution is finally transformed so that the distances between the node positions correspond to the original distances in the matrix. The outline of the algorithm using a classical metric MDS solver is as follows [58]:

1. Compute the shortest length paths between all pairs of nodes in the graph. The shortest path lengths give distance estimates between all pairs of nodes that are used to build the distance matrix for MDS.

2. Apply a classical MDS solver using the previously computed distance matrix. Keep the first 2 largest eigenvalues and eigenvectors to construct a relative map of the nodes in two dimensions.

3. Perform a transformation of the relative map to obtain the absolute positions of the nodes.

One of the benefits of this algorithm is that it is easy to analyse its computational complexity. Let n be the number of nodes in the sensor network. The first step uses the Floyd-Warshall algorithm to compute the shortest paths which has a worst case running time of $O(n^3)$. The MDS solver used in Step 2 needs to perform an eigen-decomposition which also has a worst case running time $O(n^3)$ [56]. Finally the coordinate system produced by Step 2 must be transformed to match the coordinate system defined by the seed nodes. The parameters of the transformation takes

$O(m^3)$ time to compute where m is the number of seed nodes. This step may be omitted if the network is without seed nodes.

MDS-MAP is an admissible, albeit centralized, solution to the positioning problem under the defined model. Although the algorithm assumes the graph is known prior to the beginning of the algorithm which is prohibited by the defined model, the graph could be obtained by sequence of message exchanges between the nodes so that the entire graph is eventually known to a single node. Once the solution is computed at the node, the corresponding position assignments are distributed to all the nodes.

While the MDS-MAP algorithm is a valid approach, is has several drawbacks. The fundamental issue is that its message complexity is high because of the additional step required to describe the graph at a single node before processing. Another critical issue with the algorithm is that it is somewhat computationally expensive to run on resource constrained devices. In addition to the aforementioned computational issues the algorithm does not produce good position assignments in irregular embeddings [57].

To address these issues some authors have suggested a decentralized approach where node positions are computed locally using an MDS solver and then glued together into a global coordinate system [26, 56, 57].

MDS-MAP(P) Algorithm Outline:

1. Compute a local coordinate system at each node u_i:

   (a) Compute the shortest paths between all pairs of nodes that are a graph distance $h_{\max}$ from node u_i. The shortest path lengths are used as the entries in the distance matrix.

   (b) Use MDS with the distance matrix computed in the previous step and keep the first two largest eigenvalues and eigenvectors to construct the local map in two dimensions.

2. Merge the local maps into clusters with consistent coordinate systems

3. Transform the relative node positions to absolute positions using the seed nodes.

The key difference between MDS-MAP and MDS-MAP(P) is that the MDS operation is executed on each node to compute local coordinate systems and these local coordinate systems need to be merged to form a global coordinate system. Steps 1(a) and 1(b) of MDS-MAP(P) are identical to those of MDS-MAP and share the same time complexity. Therefore, if k is the average neighbourhood size, then Step 1 has a running time of $O(k^3)$ at each node. Beginning at a random node, the local coordinate systems are merged in a greedy manner. During each merge operation the conformation difference between the two coordinate systems is minimized by finding the best linear transformation. The merge phase of the

algorithm takes $O(k^3n)$ time. Finally a transformation to the coordinate system defined by the seed nodes is computed, taking $O(m^3)$ time.

The authors also give the message complexity of their algorithm, which is dependent on the size of the local coordinate systems. To construct the distance matrix a node must learn its second neighbours. After each node computes a local coordinate system they must be merged to a global coordinate system. With a binary aggregation tree each local coordinate system propagates over $O(\log n)$ hops making the total communication cost $O(n \log n)$.

Figure 3.6 gives a typical connectivity graph input to these algorithms. Figure 3.7 gives the mean positional accuracy of MDS-MAP(P) and DV-Hop when the nodes are randomly embedded in bounded regions of $\mathbb{R}^2$.

The benefit of MDS-based algorithms is that they provide good quality embeddings with just three seed nodes or even none at all. Unfortunately, for resource limited sensor devices, the computational complexity of MDS is relatively high due largely to the eigen-value decompositon. If the number of seed nodes is proportional to the total number of nodes the merging phase would be quite minimal and could possibly be completely removed since information about the global coordinate system is locally obtainable.

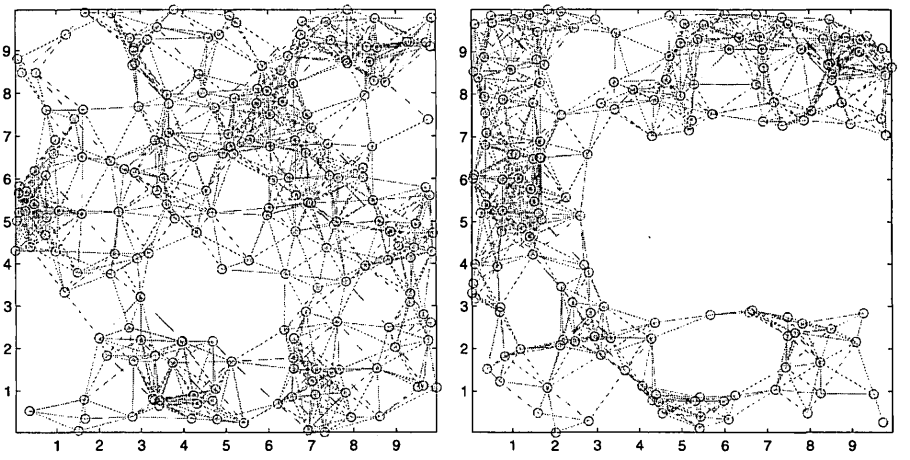


Figure 3.6: An example of a network used as input to the algorithm MDS-MAP(P) and DV-Hop [57]. The network on the left was induced by nodes embedded in a rectangular shaped space while the network on the right was induced by nodes embedded in a C-shaped space.

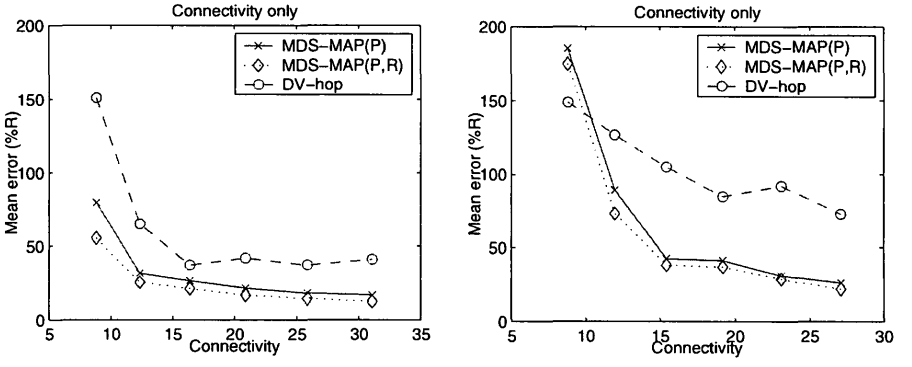


Figure 3.7: The Expected Positional Error: MDS-MAP(P) and DV-Hop [57]. The graph on the left is the result of running the algorithms on a 200 node network, four of which were seed nodes, where the nodes are embedded in a rectangular shaped space. The graph on the right is the result of running the algorithms on a 160 node network, four of which were seed nodes, where the nodes are embedded in a C-shaped shaped space. The node positions are completely random within the environment.

3.4 Sampling-based Algorithms

The sampling-based algorithms represent a node's position as a discrete probability distribution that is described by a set of sample points. The connectivity information is used to compose a region of possible positions for a node and each sample point in the region has some probability of being the true position of the node.

All algorithms to be described are designed to be executed locally at the node and have three main steps: (i) create samples of possible positions from $\mathbb{R}^2$, (ii) filter the positions based on known constraint information, (iii) compute an estimated position q_i as the average of all remaining samples.

The advantage of sample-based schemes is that they are incremental, making them adaptive to changes in the node embedding due to node movement.

The first researchers to propose Monte Carlo sampling techniques for positioning in sensor networks were Hu and Evans with their MCL algorithm [21]. Their method guesses some possible positions for a node and then weights the sample positions with a zero or one in accordance to their agreement with constraints. The low weight position estimates are discarded and replaced by new sample positions.

Initially, at time step $t = 0$, there are no previous samples so the algorithm samples from the deployment area. The algorithm maintains a set of sample points L_t. In each time step t, the algorithm takes each sample in $l_{t-1}^k \in L_{t-1}$, $j \in$

$\{1, \ldots, \eta\}$, from time step $t-1$ and generates a new sample l_t^k that is a completely random point in the disk $\mathcal{D}(l_{t-1}^k, \delta)$. The new sample positions are then filtered based on their consistency with constraint information imposed by neighbour seed node and second neighbour seed nodes. A constraint based on a neighbour seed node u_j requires the new sample l_t^k to satisfy the condition $d(p_t(u_j), l_t^k) \leq r$. A constraint based on a second neighbour seed node u_j requires the sample l_t^k to satisfy the condition $r < d(p_t(u_j), l_t^k) \leq 2 \cdot r$ since these nodes are not neighbours of u_i but they are neighbours of u_j's neighbours.

To produce good position estimates, the minimum number of samples η has been set to 50, which was determined empirically [21]. Since the filtering step removes samples that are inconsistent with the constraints, the sample generation step is repeated until η consistent samples have been accumulated. Finally, the average of the candidate sample positions is computed

$$q_t(u_i) = \frac{\sum_{k=1}^{\eta} l_t^k}{|L_t|} \tag{3.1}$$

and assigned as the estimated position of node u_i.

MCL Algorithm Outline:

1. Initialize the set of sample points L_0 to be η completely random points from the deployment region

2. For each time step t, compute a new set of sample points L_t based on the previous possibilities L_{t-1}:

   (a) $L_t = \emptyset$

   (b) While $|L_t| < \eta$

      i. For each possible position $l_{t-1}^k \in L_{t-1}$, uniformly sample a new position l_t^k from the disk $\mathcal{D}(l_{t-1}^k, \delta)$

      ii. If the sample position l_t^k satisfies the constraints imposed by the neighbour seed nodes and the second neighbour seed nodes, then add it to sample set L_t

   (c) Compute an estimated position $q_t(u_i) = \frac{\sum_{k=1}^{\eta} l_t^k}{|L_t|}$

The immediate drawback of MCL is that it requires a large number of seed nodes in comparison to other hop-based algorithms like DV-Hop. The authors observe that when the node speed δ is low relative to the communication range r, positional accuracy can increase. This observation is explained by an effective increase in the seed node density due to mobility.

On the other hand, MCL is wasteful as it does not make use of all available constraint information. This drawback is addressed in the algorithm MSL* [53].

The algorithm MSL* generalizes MCL by considering constraints imposed by non-seed nodes in addition to the those imposed by seed nodes. The constraints

imposed by non-seed nodes must be adjusted by δ because they are based on the sample positions of the node which are created in the previous time step.

The first step of this algorithm is similar to MCL: a node generates new samples at time step t which are based on those samples from time step $t-1$.

Although not explicit, the samples used in MCL are dichotomous. A sample position in MCL that satisfies all the constraints has weight one and otherwise it has a weight of zero. Since the assigned positions of non-seed nodes have positional error, using constraints imposed by non-seed nodes can introduce robustness issues. Therefore a partial weight is given to a sample position which is a measure of how consistent it is with the constraints imposed by an seed or non-seed node. The product of the partial weights corresponding to constraint satisfaction gives the total weight of a sample. A sample position is kept if it has some minimum weight.

The final phase of MSL* is to randomly copy η samples from the current sample set into a new set, selecting each new sample with a probability proportional to its weight. This process tends to replace the lower weighted samples with higher weighted samples.

The authors identify the use of the sample positions of a node's neighbours as potentially costly due to large message sizes, and propose an alternative algorithm called MSL. This variation of MSL* has nodes passing their position estimates, a composition of the samples, as opposed to the samples themselves. To gauge the

quality of the position estimate a node also passes its estimated accuracy of its position calculation.

Both the MCL and MSL* algorithms spend a considerable amount of time finding candidate sample positions – sample positions that are consistent with constraints. Since the constraints are known before generating sample positions, one idea to improve the sampling efficiency of these algorithms is to create a bounding box based on constraint information from seed nodes [4]. Figure 3.8 illustrates a set of sample points contained in a bounding box.

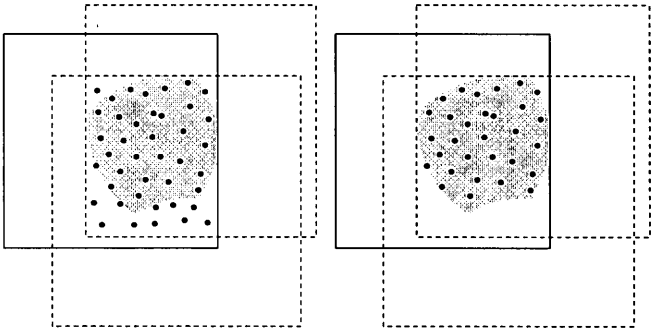


Figure 3.8: The region of feasible positions for a node is shaded grey. The illustration on the left shows a set of sample positions for a node taken from the intersection of the bounding boxes. On the right, sample positions that do not satisfy the constraints are removed so that only feasible positions for the node are remaining. The solid rectangles are boxes derived from constraints imposed by seed nodes and the dashed rectangles are boxes derived from constraints imposed by non-seed nodes. The constraints imposed by non-seed nodes are adjusted to account for positional error.

Samples are then drawn from a bounding box constructed from constraints, making it more likely the sample positions are consistent with the constraints. The coordinates $(x_{min}, x_{max}, y_{min}, y_{max})$ of the bounding box constructed with neighbour seed nodes are

$$\begin{aligned} x_{\min} &= \max_{j=1}^{\dot{N}_t(u_i,1)} \{x_{p_t}(u_j) - r\} \\ x_{\max} &= \min_{j=1}^{\dot{N}_t(u_i,1)} \{x_{p_t}(u_j) + r\} \\ y_{\min} &= \max_{j=1}^{\dot{N}_t(u_i,1)} \{y_{p_t}(u_j) - r\} \\ y_{\max} &= \min_{j=1}^{\dot{N}_t(u_i,1)} \{y_{p_t}(u_j) + r\}, \end{aligned}$$

where $\dot{N}_t(u_i, 1)$ is the set of first neighbour seed nodes of u_i and $p_t(u_j) = (x_{p_t}(u_j), y_{p_t}(u_j))$ is the position of seed node $u_j \in \dot{N}_t(u_i, 1)$ at time t [66, 4]. Any second neighbour seed nodes can be included by using $2 \cdot r$ instead of r for the bound on distance.

For each sample in $l_{t-1}^j \in L_{t-1}$, a new sample is drawn from the intersection of the square box with sides of length $2 \cdot \delta$ and the aforementioned bounding box. The sample position is then kept if it satisfies the constraints. This process is repeated until there are η valid samples. Notice if there are no seed nodes then there are no new constraints to restrict a newly generated sample position.

A further refinement of the bounding box idea is proposed by Zhang et al. [66,

67]. The authors explain that the size of the bounding box can be further reduced by including more negative constraint information, and using constraints imposed by non-seed nodes.

Negative constraints take the form of geometric regions where a node position cannot be located. This type of constraint is used in the previously described algorithms but it can also be used to restrict the size of the bounding box as illustrated in Figure 3.9.

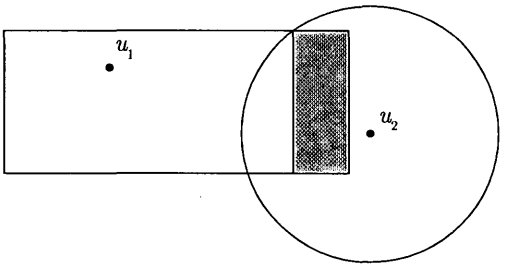


Figure 3.9: Reducing the size of a bounding box for node u_1 using the negative information imposed by the non-neighbour node u_2. The shaded portion of the whole rectangle can be removed to produce a smaller bounding box since this region is entirely enclosed inside the communication region of node u_2.

Constraints derived from non-seed nodes can also reduce the size of the bound box. After constructing the bounding box using information from seed nodes, the authors suggest it be updated using the non-seed neighbours $N_t(u_i, 1)$ of node u_i

$$x_{\min} = \max\{x_{\min}, \max_{u_j \in N_t(u_i,1)} \{x_{q_{t-1}}(u_j) - \delta - r - e_{x_{q_{t-1}}}(u_j)\}\}, \tag{3.2}$$

and node u_i's position at time $t-1$

$$x_{\min} = \max\{x_{\min}, x_{q_{t-1}}(u_i) - \delta - e_{x_{q_{t-1}}}(u_i)\}, \tag{3.3}$$

where $q_{t-1}(u_i) = (x_{q_{t-1}}(u_i), y_{q_{t-1}}(u_i))$ is node u_i's estimated position at time $t-1$ and $e_{x_{q_{t-1}}}(u_i)$, $e_{y_{q_{t-1}}}(u_i)$ is the maximum error of node u_i's estimated position at time $t-1$ for each dimension. The expressions for $x_{\max}$, $y_{\min}$, and $y_{\max}$ are analogous to those for $x_{\min}$. Like the MSL algorithm, neighbour nodes pass their estimated position and estimated positional error as opposed to the set of samples.

A recent contribution by Mirebrahim and Dehghan reveals some of the merits in removing the randomization from the sample generation process [40]. The authors suggest a deterministic approach that samples the bounding box uniformly, which removes the possibility of over sampling some regions of the bounding box. Indeed this modification is in accordance with the assumption that each point in the region of uncertainty is equally likely. Additionally the more uniform sample generation improves the accuracy of the estimated maximum positional error. The distance between samples is experimentally determined by a computer simulation.

An alternative representation of the feasible set of positions was proposed by Datta et al. [13]. In their algorithm the boundary of the feasible set is described by a convex polygon, rather than using an approximation of the entire set of feasible

positions for a node. Nodes exchange messages containing the polygon representing their region of candidate positions. To compute its own candidacy region, a node performs dilation and intersection operations on received polygons and reports the centroid of the resulting polygon as its estimated position.

3.5 Probabilistic Analysis of Connectivity-based Positioning

In this section we describe approaches to bounds on the positional accuracy of connectivity-based positioning algorithms. There are two notable bounds relating to the quality of position estimates under the Unit Disk communication model. Each is probabilistic, with one giving the expected size of the of the set of possible positions for a node, and the other giving a bound on the expected achievable accuracy of a positioning algorithm. Both analyses assume that each point in the space containing the nodes is an equally likely position for a node.

Sidi et al. consider a set of node nodes placed randomly within the unit disk $\mathbb{D}$ in $\mathbb{R}^2$, each with a communication radius of r [61]. The authors only consider constraints from E that are imposed directly by seed nodes.

The **region of uncertainty** $\mathcal{R}(u_i)$ for a typical node u_i is the subset of $\mathbb{D}$ that contains all feasible positions for a node that satisfy the constraint information used by an algorithm,

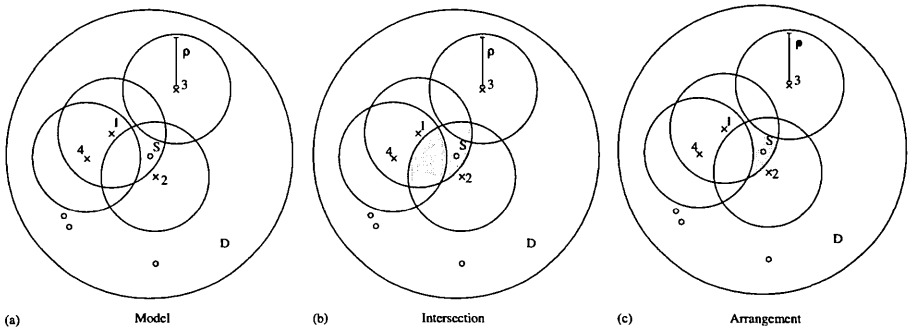


Figure 3.10: Figure (a) gives an illustration of a node configuration where each disk represents the communication region of a node. Figure (b) gives the region of uncertainty of a node where only positive constraints are used while Figure (c) gives the region of uncertainty where both positive and negative information is used to compose the region of uncertainty [61].

$$\mathcal{R}(u_i) = \begin{cases} \bigcap_{u_j \in \dot{N}(u_i,1)} \mathcal{D}(p(u_j), r) \setminus \bigcup_{u_j \in \dot{N}^c(u_i,1)} \mathcal{D}(p(u_j), r) & : \dot{N}^c(u_i, 1) \neq \emptyset \\ \mathbb{D}_r \setminus \bigcup_{u_j \in \dot{N}^c(u_i,1)} \mathcal{D}(p(u_j), r) & : \text{otherwise} \end{cases}$$

Here $\dot{N}(u_i, 1)$ is the set of seed nodes that are first neighbours of node u_i and $\dot{N}^c(u_i, 1)$ is the set of seed nodes that are not first neighbours of node u_i. To avoid boundary issues only nodes within the region $\mathbb{D}_r = \mathcal{D}(1 - r)$ are considered. The expectation of X, the size of the region of uncertainty $\mathcal{R}(u_i)$ of node u_i, is

$$\mathbb{E}(X|u_i \in \mathbb{D}_{3r}) = \int_{\mathbb{D}_r \setminus \mathcal{D}(u_i,2r)} \mathbb{P}[x \in \mathcal{R}(u_i)|E_1]dx + \int_{\mathcal{D}(u_i,2r)} \mathbb{P}[x \in \mathcal{R}(u_i)|E_2]dx, \tag{3.4}$$

where E_1 is the event that $x \notin \mathcal{D}(p(u_i), 2r)$ and E_2 is the event that $x \in \mathcal{D}(p(u_i), 2r)$.

Due to the uncertainty in the node positions, the authors compute the size of the region by finding the probability that no seed neighbours nodes reside in the region $\mathcal{D}(p(u_i), r) \cup \mathcal{D}(x, r) \setminus \mathcal{D}(p(u_i), r) \cap \mathcal{D}(x, r)$. Expressed in a slightly different way, the point x falls in the region of uncertainty if all seed nodes reside in the region $\mathcal{D}(p(u_i), r) \cap \mathcal{D}(x, r)$ (a neighbour or second neighbour seed node) or outside of the region $\mathcal{D}(p(u_i), r) \cup \mathcal{D}(x, r)$.

Nagpal et al. give a bound on the expected best accuracy of a positioning algorithm by considering the slackness of the constraints [44]. Assuming that the node positions in $\mathbb{R}^2$ are Poisson distributed with intensity λ, and the Unit Disk communication model, the authors reason by way of indistinguishable embeddings. Given an embedding, if a typical non-seed node is moved to a new position, producing a new embedding, and the connectivity graph induced from the node positions remains unchanged, then an algorithm cannot distinguish between the two embeddings. An algorithm must produce the same node position assignments for each

embedding. Figure 3.11 illustrates the movement of a node by a distance z and the regions that cannot contain local node positions without changing the connectivity graph.

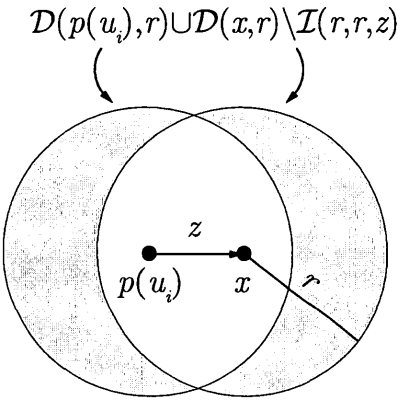


Figure 3.11: Movement of a typical node u_i a distance z in a random direction. The disk $\mathcal{D}(p(u_i), r)$ about node u_i is the communication region of the node. If the node moves to point x, then the shaded region $\mathcal{D}(p(u_i), r) \cup \mathcal{D}(x, r) \setminus \mathcal{I}(r, r, z)$ cannot contain any neighbour or non-neighbour positions since the connectivity of the graph would be changed. If a node position is contained in the left shaded region then node u_i will lose a neighbour due after moving a distance z. On the other hand, if the right shaded region contains a node position then node u_i will gain a neighbour due to the movement [3].

The largest expected distance a node can be moved before the connectivity graph changes gives a bound on positional accuracy of an algorithm. The authors compute this expected distance

$$\frac{1}{4r\lambda}, \tag{3.5}$$

where r is the communication range and λ is the density of nodes.

Chapter 4

The Positional Error Inherent to a Model

Chapter 3 presented some connectivity-based positioning algorithms that are designed for the case of mobile nodes. While node mobility presents a challenge for some algorithms, an opportunity exists for adding positional information over time which can be used to reduce positional error. This technique has been utilized in many connectivity-based positioning algorithms for mobile systems and the error reduction from the additional information has been demonstrated via computer simulation [21, 13, 53, 67]. These results prompt two questions: (i) what is the minimum achievable positional error of any connectivity-based positioning algorithm, conditioned on some positional information and, (ii) is there a limit on the usefulness of adding past positional information? This chapter addresses these questions for the Unit Disk communication model and the Random Walk mobility model

which are defined in Chapter 2. These models provide the standard framework for developing a positioning algorithm and analyzing their positional error [1].

A discrete time system is used to model the state of the system of nodes. The variable t represents the time index of a particular state. It is conventional to start the system at time index $t = 0$, which is the initial state, and increase the index with the progression of the system into the *future* so that $t + 1$ is the time immediately *after* time t. The work in this chapter breaks with convention. This chapter investigates the use of positional information from the current state and past states of the system. The current state of the system is defined to be at time index $t = 0$ and the index increases into the *past* so that the state at time index t is the time immediately *before* $t + 1$. This convention is adopted to simplify the notation and applies only to this chapter.

4.1 Positional Error

The node movement described by the model generates a sequence of unit disk graphs, which are the input to a connectivity based positioning algorithm. The constraints imposed by node connectivity are bounds on the Euclidean distance which implies that the positions of the nodes cannot always be determined exactly i.e., there is positional error. A component of positional error which applies to

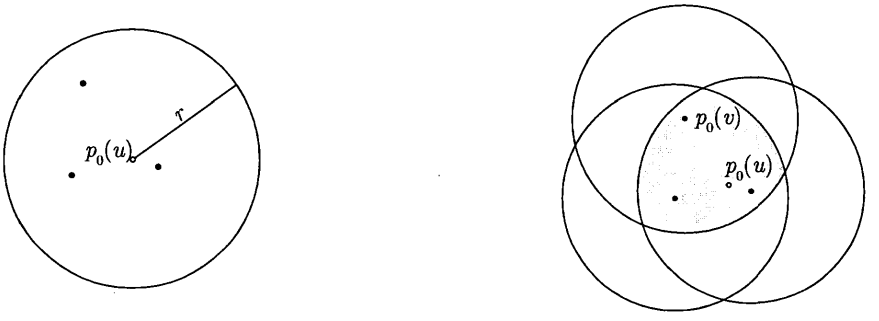
[1]This work has been published in [35, 37].

all connectivity-based algorithms operating with the unit disk and random walk models is formulated in this section.

The component of positional error inherent to model is formulated and then used to derive a lower bound on expected positional error. This section describes the model inherent error for both stationary and mobile node systems and explains how the inherent error can be measured. The description considers only constraints imposed by the neighbours of a node. In Section 4.2, the model inherent error is formulated with constraints imposed by both the neighbours and non-neighbours of a node. The non-neighbours can be discovered by communication with neighbours.

4.1.1 Stationary Node Case

Each neighbour node $v \in N_0(u, 1)$ and its position $p_0(v)$ imposes a constraint on the position of node u, where the subscript 0 refers to the current time $t = 0$. If the positions of the neighbours $v \in N_0(u, 1)$ are known, then based on the constraints, the set of feasible positions for node u is restricted to being within a distance r of the position of each neighbour. Geometrically, each positional constraint takes the form of a disk $\mathcal{D}(p_0(v), r)$ of radius r about the position of a neighbour v. The set of feasible position assignments for node u is called the **region of uncertainty** $\mathcal{R}(u)$ [61]. The node u can be assigned any point in this region and still be within a distance r of the position of each of its neighbours. A region of uncertainty for a



(a) The neighbourhood of a typical node u.

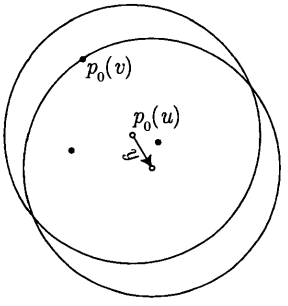
(b) Region of uncertainty for a node u derived from the positions of neighbours in the neighbourhood.

Figure 4.1: Positional constraints from the neighbourhood of a node u.

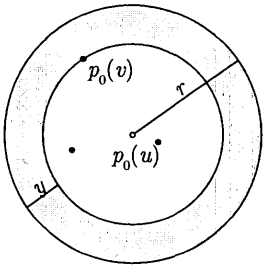
particular neighbourhood configuration is illustrated in Figure 4.1

Given the positional uncertainty, there is a maximum distance y that node u can be moved from its current position, in any direction, and still satisfy the positional constraints. Alternatively, the constraints can be perturbed by a distance y and still admit the position for node u as feasible. The distance y is called the **perturbation distance** for node u. Informally the perturbation distance is the amount of slack permitted by the positional constraints. Figure 4.2a shows a perturbation of a node u to a new position that still satisfies each of the positional constraints. The expected perturbation distance will be used to give a lower bound on the expected positional error incurred by any connectivity-based algorithm.

The expected perturbation distance is a lower bound on the expected positional error. The Poisson model generates a set of embeddings of the nodes in $\mathbb{R}^2$ and a



(a) An example of changing the position of a node u by a distance y while maintaining a distance of at most r to each neighbour.



(b) The annular region of width y, shaded grey, about the position of a node u.

Figure 4.2: Positional constraints from the neighbourhood of a node u.

probability distribution over the embeddings. The region of uncertainty is a random variable R which is a subset of $\mathbb{R}^2$ derived from an embedding and a communication model. The Poisson model is homogeneous with intensity λ and the communication model is the Unit Disk model which is described in Chapter 2.

Given a region of uncertainty $R = \mathcal{R}$, the number of points in $\mathcal{R}$ is a random variable, denoted by $X(\mathcal{R}) \sim \text{Poisson}(\lambda)$. Suppose $\mathcal{R}$ contains a single point, namely the true position of node u_i, then for any $\mathcal{B} \subseteq \mathcal{R}$, $\mathbb{P}[X(\mathcal{B}) = 1 | X(\mathcal{R}) = 1] = \frac{\mu(\mathcal{B})}{\mu(\mathcal{R})}$. That is, the position of node u_i is uniformly distributed in $\mathcal{R}$. The position estimate $q(u_i)$ of node u_i is a fixed point in $\mathbb{R}^2$.

The positional error of the position estimate $q(u_i)$ is the Euclidean distance between $q(u_i)$ and $p(u_i)$. The position $p(u_i)$ is random so the positional error is a random variable $E|R$, where R is the given region of uncertainty.

Let the boundary of $\mathcal{R}$ be denoted $\text{br}(\mathcal{R})$. The perturbation distance is the smallest distance from $p(u_i)$ to a point in $\text{br}(\mathcal{R})$. The position $p(u_i)$ is random so the perturbation distance is a random variable $Z|R$, where R is the given region of uncertainty.

It will be shown that $\mathbb{E}[Z|R] \leq \mathbb{E}[E|R]$ and therefore, by the law of iterated expectation, $\mathbb{E}[Z] \leq \mathbb{E}[E]$.

Consider a randomly selected point $p(u_i) \in \mathcal{R}$ and the line from $q(u_i)$ through $p(u_i)$ to the nearest boundary point $p^* \in \text{br}(\mathcal{R})$ that is at or beyond $p(u_i)$.

There are two types of lines from $q(u_i)$:

1. The lines $L_1 \subseteq L$ from q to p^* that intersect one boundary point of $\mathcal{R}$, namely p^*. Let $\mathcal{R}_1 = \{p(u_i)|p(u_i) \in \bigcup_{\mathcal{L} \in L_1}^{\infty} \mathcal{L} \cap \mathcal{R}\ \}$.

2. The lines $L_2 \subseteq L$ from $q(u_i)$ to p^* that intersects more than one boundary point of $\mathcal{R}$. Let $\mathcal{R}_2 = \mathcal{R} \setminus \mathcal{R}_1$.

Figure 4.3 illustrates the two types of lines.

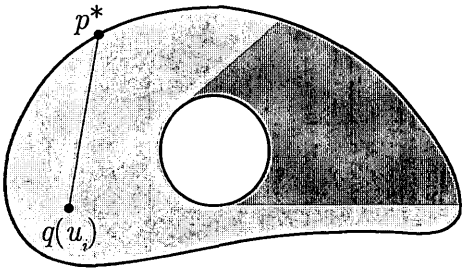


Figure 4.3: A region of uncertainty $\mathcal{R}$ with a hole. The points in $\mathcal{R}$ are shaded grey and the boundary of $\mathcal{R}$ is black. The points in $\mathcal{R}_1$ are shaded light grey and the points in $\mathcal{R}_2$ are shaded dark grey. The point labelled $q(u_i)$ is the estimated position of the node u_i.

The two different types of lines leads to two different cases for the lower bound on the expected error.

Case 1: Consider the set of lines L_1. Bisect each of the lines so that an inner region A and outer region B are created. It is easy to see that $\mu(A) < \mu(B)$ and therefore $\mathbb{P}[p(u_i) \in A] < \mathbb{P}[p(u_i) \in B]$ by the uniformity of $p(u_i)$ in $\mathcal{R}$. Let random variable $E|R$ be the distance from q to $p(u_i)$, random variable $F|R$ be the distance from $p(u_i)$ to p^*. We have $\mathbb{E}[F|R] < \mathbb{E}[E|R]$ since the expected position is in B, which is more than half the distance to the boundary $\mathrm{br}(\mathcal{R})$. The perturbation distance $Z|R$ is the smallest distance from $p(u_i)$ to a point in $\mathrm{br}(\mathcal{R})$ so $\mathbb{E}[Z|R] \leq \mathbb{E}[F|R]$.

Case 2: Consider the set of lines L_2 and let p' be the boundary point of $\mathcal{R}$ on $\mathcal{L} \in L_2$ that is closest to $q(u_i)$. The positional error is $d(q(u_i), p') + d(p', p(u_i))$ but

the perturbation distance is at most $d(p', p(u_i))$ so $\mathbb{E}[Z|R] \leq \mathbb{E}[E|R]$.

Therefore, the inequality $\mathbb{E}[Z|R] \leq \mathbb{E}[E|R]$ holds in both cases and taking the marginal over R we have $\mathbb{E}[Z] \leq \mathbb{E}[E]$.

The calculation of the perturbation distance y for node u is the focus of the analysis. To facilitate this calculation we introduce the **annular region**: a region of a two-dimensional Euclidean space that is the set difference of two disks about the position of a node u_i. The width of an annular region is the difference in the radii of the two disks used in its construction. If the width is y, then the annular region about $p(u)$ is denoted by $\mathcal{A}_0(p(u), y)$. When the position of node u is clear from the context, the centre of the annular region will be omitted from the notation and will be written $\mathcal{A}_0(y)$. The annular region is illustrated in Figure 4.2b. We are interested in the maximum width of an annular region about the position of a node u that does not contain any neighbour positions. If the position of a neighbour $v \in N_0(u, 1)$ is contained in the annular region of width y, then the node u can be moved a distance y away from its current position so that it is a distance greater than r away from the position of node v. This new position, after the movement, would violate the constraints on the position of node u. Therefore to determine the perturbation distance it is sufficient to find the largest annular region that does not contain any of the neighbour positions. Note the perturbation of a node does not correspond to a physical movement but rather a hypothetical movement to quantify

the slackness of the positional constraints.

Each candidate position in the region of uncertainty has a perturbation distance. From the model there is a probability associated with each position, so the perturbation distance is a random variable denoted by Y_0. This random variable represents the distance that a randomly selected node can be perturbed from its position and not violate any of the positional constraints. Consider a bounded set $\mathcal{X} \subset \mathbb{R}^2$ and let $X_0(\mathcal{X})$ be the number of neighbour node positions in the set $\mathcal{X}$. The event of at least one neighbour position occurring in the annular region of width y about the point $p(u)$ is examined. Since the perturbation distance Y_0 is the distance a node can be perturbed without violating a constraint, we have $(X_0(\mathcal{A}_0(y)) > 0))$ implying $Y_0 \leq y$. The event of at least one neighbour position in the annular region has probability

$$\mathbb{P}[X_0(\mathcal{A}_0(y)) > 0] = 1 - \mathbb{P}[X_0(\mathcal{A}_0(y)) = 0]. \tag{4.1}$$

So the probability distribution of Y_0 can be computed from the probability of an empty annular region

$$F_{Y_0}(y) = \mathbb{P}[Y_0 \leq y] = 1 - \mathbb{P}[X_0(\mathcal{A}_0(y)) = 0]. \tag{4.2}$$

The expected perturbation distance, denoted $\mathbb{E}[Y_0]$, is a lower bound on the

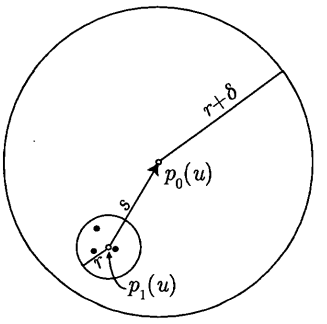
expected positional error incurred by any connectivity-based positioning algorithm since any point within a distance $\mathbb{E}[Y_0]$ of the true, albeit unknown, position of a node is expected to satisfy all of the constraints.

4.1.2 Mobile Node Case

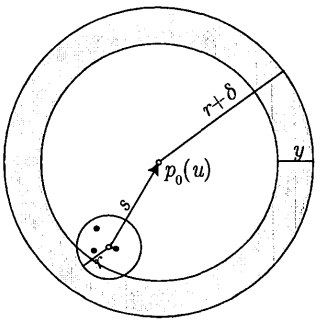
The above analysis of perturbation distance is for the current time step $t = 0$. Suppose the node is physically moved a distance δ in one time step. The connectivity information from this step is now associated with time step $t = 1$. Using constraints from the two time steps, the distance between the new position of node u and its neighbours is now less than or equal to $r + \delta$. Figure 4.4a illustrates a physical movement of a node u, with each of its neighbours, encountered before the move, a distance of at most $r + \delta$ from its new position.

After the movement node u may no longer be neighbours with some of the nodes in the set $N_1(u, 1)$. This change requires an adjustment to preserve the validity of the positional constraints. Adopting the previous strategy, the annular region about the current position of node u is considered. Since the positional constraints now impose a limit of $r+\delta$ on the distance between the new position of node u and each of the nodes in the set $N_1(u, 1)$, the annular region has been adjusted accordingly as shown in Figure 4.4b.

After the physical move node u encounters a new neighbourhood at its new



(a) The position of node u changed between time $t = 1$ and time $t = 0$. The node moved a distance $s \leq \delta$. The neighbourhood $N_1(u, 1)$ of node u at time $t = 1$ is illustrated as the disk about the point $p_1(u)$)

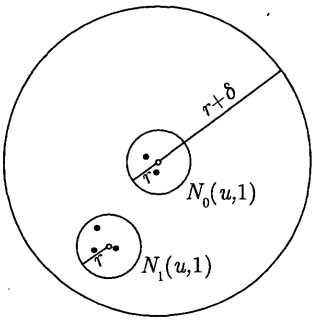


(b) The annular region of width y about the position of node u at time $t = 0$. All neighbours of node u at time $t = 1$ must be within a distance $r + \delta$ of point $p_0(u)$.

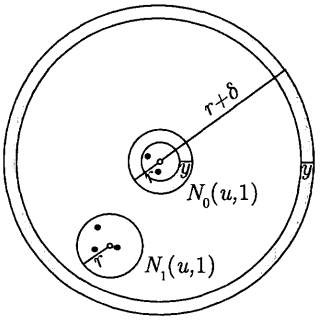
Figure 4.4: Perturbation of node u that maintains the positional constraint between the position of node u and the position of each of its neighbours.

position. Let the neighbourhood of node u from the last step be $N_1(u, 1)$ and the neighbourhood of node u from the current step be $N_0(u, 1)$. The two neighbourhoods are shown in Figure 4.5a.

There are now two sets of positional constraints, one for each neighbourhood. One set of constraints requires the current position of node u to be within a distance $r + \delta$ of the positions of the nodes in the set $N_1(u, 1)$. The other set of constraints requires the current position of node u to be within a distance r of the positions of the nodes in the set $N_0(u, 1)$. To find the maximum perturbation distance y when the two sets of positional constraints are considered, an annular region is used for each set of constraints as illustrated in Figure 4.5b.



(a) The introduction of a second neighbourhood, $N_0(u,1)$, after the physical movement of node u.



(b) Two empty annular regions: one for the positional constraints for time step $t = 1$ and the other for the positional constraints for time step $t = 0$.

Figure 4.5: The introduction of a second neighbourhood produces two sets of positional constraints.

Consider the annular regions about node u, $\mathcal{A}_1(y)$ and $\mathcal{A}_0(y)$ from the previous and current time steps, respectively. Let $X_1(\mathcal{A}_1(y))$ be the number of nodes from $N_1(u,1)$ with positions in the annular region of width y: $\mathcal{A}_1(y)$ for time step 1, and $X_0(\mathcal{A}_0(y))$ be the number of nodes from $N_0(u,1)$ with positions in the annular region of width y: $\mathcal{A}_0(y)$ for time step 0. The probability of the event of no node positions in either of the regions is

$$\mathbb{P}[X_0(\mathcal{A}_0(y)) = 0, X_1(\mathcal{A}_1(y)) = 0]. \tag{4.3}$$

Then the probability that at least one of the annular regions of width y has at least one node position is

$$\mathbb{P}[X_0(\mathcal{A}_0(y)) + X_1(\mathcal{A}_1(y)) > 0] = 1 - \mathbb{P}[X_0(\mathcal{A}_0(y)) = 0, X_1(\mathcal{A}_1(y)) = 0]. \quad (4.4)$$

If random variables Y_0 and Y_1 are the perturbation distances corresponding to the constraints from time step 0 and time step 1, respectively, Equation 4.4 is the minimum of Y_0 and Y_1. If $Z_1 = \min\{Y_0, Y_1\}$ then $X_0(\mathcal{A}_0(y)) + X_1(\mathcal{A}_1(y)) > 0$ implies $Z_1 < y$, and $\mathbb{P}[Z_1 < y] = \mathbb{P}[X_0(\mathcal{A}_0(y)) + X_1(\mathcal{A}_1(y)) > 0]$.

In general, for T physical movements of node u to reach its current position, the probability that at least one of the annular regions of width y has at least one node position is

$$\mathbb{P}[X_0(\mathcal{A}_0(y)) + \cdots + X_T(\mathcal{A}_T(y)) > 0)] = 1 - \mathbb{P}[X_0(\mathcal{A}_0(y)) = 0, \ldots, X_T(\mathcal{A}_T(y)) = 0], \quad (4.5)$$

where the random variable $X_t(\mathcal{A}_t(y))$ is the number of positions of nodes from the set $N_t(u, 1)$ at time t (t time steps before the current time) contained in the annular region $\mathcal{A}_t(y)$. This generates a probability distribution over values of z, the width of the annular region $\mathcal{A}_t(z)$ for all time steps t. Let random variable Z_T be the minimum width of all annular regions where at least one annular region contains

the position of a neighbour:

$$Z_T = \min\{Y_0, \ldots, Y_T\}.$$

Then $P[Z_T < z] = P[X_0(\mathcal{A}_0(z)) + \cdots + X_T(\mathcal{A}_T(z)) > 0]$, and the position and movement model assumptions generate a distribution for Z_T. The expected value $\mathbb{E}[Z_T]$ is a lower bound on the expected positional error of any connectivity based positioning algorithm in the case of T discrete isotropic movements of length δ to a current position. This intrinsic component of positional error is the object of interest. It will be developed explicitly and the impact of node mobility on positional error will be characterized.

4.2 Formulation of Positional Error

This section begins a comprehensive derivation of the bound on the minimum expected positional error of connectivity-based positioning algorithms in mobile systems for the model with unit disk communication and random walk mobility. Initially the expected value of Z_T is found given a specific sequence of T movements.

4.2.1 Positional Constraints

The analysis considers a system with a set of nodes V located in a two-dimensional Euclidean space $\mathbb{R}^2$ and the distance between two points p and q in $\mathbb{R}^2$, denoted $d(p, q)$, is the Euclidean distance. The nodes are capable of changing their position over time which is modelled in a discrete-time system. Each node makes an isotropic movement of length δ at time t.

The sequence of neighbourhoods $N_0(u, 1) \ldots N_t(u, 1)$ is used to impose constraints on the set of feasible positions for node u. This type of constraint is referred to as a positive positional constraint since it specifies that the point $p_0(u)$ is within some distance of another point.

Definition 1. *The positions of the neighbour nodes $v \in N_t(u, 1)$ all satisfy the inequality $d(p_0(u), p_t(v)) \leq r + t\delta$. These determine* ***positive positional constraints*** *for node u at time t. We use r_t^+ to denote the distance $r + t\delta$.*

With respect to node u, the other nodes of the system can be categorized as neighbours or non-neighbours. The non-neighbours of node u can be used to derive positional constraints of a different form than those from neighbours of u. Specifically, the minimum distance between node u and any of its non-neighbours at time $t = 0$ must be greater than r. Since movement is constrained by δ between each time step, a positional constraint from time step $t > 0$ can be preserved by reducing

the distance to the point by $t\delta$. This type of constraint is referred to as a negative positional constraint since it specifies the point $p_0(u)$ is not within some distance of another point. The inclusion of negative positional constraints can reduce the size of the region of uncertainty. A discussion of the efficacy of negative constraints in practice is presented in Section 4.3.

Definition 2. *The positions of the non-neighbour nodes $w \in N_t^c(u,1)$ all satisfy the inequality $d(p_0(u), p_t(w)) > r - t\delta$ where $t\delta \leq r$ and 0 otherwise.. These determine* ***negative positional constraints*** *for node u at time t. We use r_t^- to denote the distance $r - t\delta$.*

Recall that the region of uncertainty $\mathcal{R}(u)$ is the set of feasible positions for a node that is defined by the positional constraints [61]. Geometrically this is the set of points contained in the intersection of the disks formed by positive constraints minus the union of disks formed by negative constraints.

4.2.2 Composition of Annular Regions

We would like to find the perturbation distance z from point $p_0(u)$ so that neither the positive constraints nor the negative constraints from any time step are violated. The annular region previously described is adjusted to accommodate the inclusion of both the positive and negative constraints by having an inner annular region and an outer annular region. The inner annular region at time t is the set $\mathcal{A}_t^+(p, y_t) =$

$\mathcal{D}(p, r_t^+) \setminus \mathcal{D}(p, r_t^+ - y_t)$ and the outer annular region is the set $\mathcal{A}_t^-(p, y_t) = \mathcal{D}(p, r_t^- + y_t) \setminus \mathcal{D}(p, r_t^-)$. The modified annular region is shown in Figure 4.6. Extending the reasoning from Section 4.1, a perturbation of distance y_t does not violate any negative positional constraints at time t if the outer annular region of width y_t is empty and just as before, this perturbation distance does not violate any positive positional constraints if the inner annular region of the same width is empty.

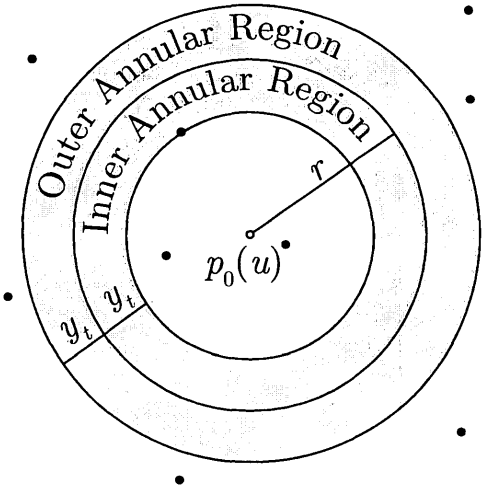


Figure 4.6: The inner and outer annular region of width y_t about the position of a node u at time step $t = 0$.

The positions of the neighbours of node u at time t are contained in the disk $\mathcal{D}(p_t(u), r)$ and therefore only a subset of the inner annular region $\mathcal{A}_t^+(p_0(u), y_t)$ cannot contain any neighbour positions from time step t. Similarly the positions of non-neighbours from time step t are not contained in the disk $\mathcal{D}(p_t(u), r)$ and therefore some subset of the outer annular region $\mathcal{A}_t^-(p_0(u), y_t)$ cannot contain any

non-neighbour positions from time step t. The Euclidean distance s_t between the position $p_t(u)$ at time t and the position $p_0(u)$ at time 0 is the **resultant** of t steps of the random walk. The regions that cannot contain any neighbour or non-neighbour positions are dependent on several variables: (i) the time step t, (ii) the resultant s_t, and (iii) the width of the annular region y_t. In this subsection the analysis is conditioned on the resultants $\{s_1, \ldots, s_t\}$ but Section 4.2.5 analyzes these values and provides an unconditioned result.

Let $\mathcal{E}_t^+(y_t|s_t) = \mathcal{A}_t^+(p_0(u), y_t) \cap \mathcal{D}(p_t(u), r)$ and $\mathcal{E}_t^-(y_t|s_t) = \mathcal{A}_t^-(p_0(u), y_t) \setminus \mathcal{D}(p_t(u), r)$. For simplicity, the centres of the disk and annular region are implied and not listed in the arguments for the sets $\mathcal{E}^+(\cdot)$ and $\mathcal{E}^-(\cdot)$. The sets $\mathcal{E}_t^+(y_t|s_t)$ and $\mathcal{E}_t^-(y_t|s_t)$ are disjoint since neighbours from time step t are strictly contained in the disk $\mathcal{D}(p_t(u), r)$ while non-neighbours are not. Collectively the region that cannot contain any neighbour or non-neighbour positions from time step t is

$$\mathcal{E}_t(y_t|s_t) = \mathcal{E}_t^+(y_t|s_t) \cup \mathcal{E}_t^-(y_t|s_t) \tag{4.6}$$

and we are interested in the probability that $\mathcal{E}_t(y_t|s_t)$ is empty.

There are many possibilities for embeddings which are consistent with the constraints and correspondingly there is a wide range of perturbation distances. As the perturbation distance increases the width of the annular region grows. In terms

of the intersection of disks, there are three phases that are described in Tables 4.1, 4.2, and 4.3, where the shaded region cannot contain the position of any nodes. Each phase is associated with one of three types of regions: the empty set, a lune or a disk. The size of each region is denoted by the function $\mu(\cdot)$. Clearly $\mu(\emptyset) = 0$ and $\mu(\mathcal{D}(r)) = \pi r^2$ for the empty set and the disk regions respectively. The asymmetric lens $\mathcal{I}(r_1, r_2, s)$ is the region formed by the intersection of two disks $\mathcal{D}(r_1)$ and $\mathcal{D}(r_2)$ that are a distance s apart and has a size [62]

$$\begin{aligned}\mu(\mathcal{I}(r_1, r_2, s)) &= r_1^2 \arccos\left(\frac{s^2 + r_1^2 - r_2^2}{2sr_1}\right) + r_2^2 \arccos\left(\frac{s^2 + r_2^2 - r_1^2}{2sr_2}\right) \\ &- \frac{1}{2}\sqrt{(-s + r_1 + r_2)(s + r_1 - r_2)(s - r_1 + r_2)(s + r_1 + r_2)}.\end{aligned}$$

The size of the asymmetric lens is used to calculate the size of a lune.

The function μ gives the size of the regions $\mathcal{E}_t^+(y_t|s_t)$ and $\mathcal{E}_t^-(y_t|s_t)$ for a particular annular width y_t. These correspond to the shaded regions illustrated in Tables 4.1, 4.2 and 4.3. They describe the accumulation of area, accounting for the different regions encountered over different values of y_t. From the definition of a neighbourhood the largest size of the set $\mathcal{E}_t^+(y_t|s_t)$ is $\mu(\mathcal{D}(r))$.

The Tables 4.1, 4.2 and 4.3 describe specific regions of space that cannot contain node positions since otherwise positional constraints will be violated.

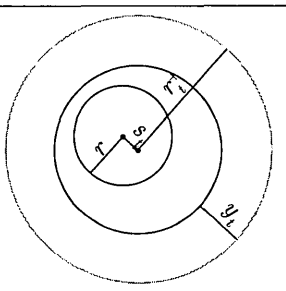
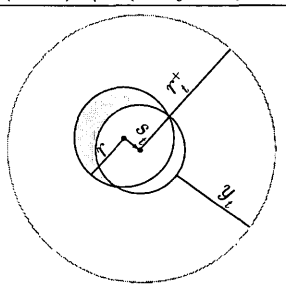
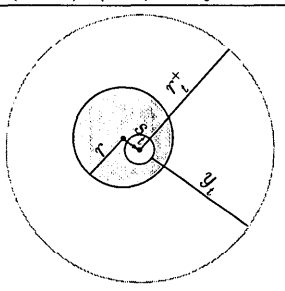
Intervals	$r_t^+ \geq r_t^+ - y_t \geq r + s_t$	$r + s_t > r_t^+ - y_t > r - s_t$	$r - s_t \geq r_t^+ - y_t \geq 0$
Region	$\emptyset$	$\mathcal{D}(p_t, r) \setminus \mathcal{I}(r, r_t^+ - y_t, s_t)$	$\mathcal{D}(p_t, r) \setminus \mathcal{D}(p_0, r_t^+ - y_t)$
Illustration			

Table 4.1: The intervals for $r_t^+ - y_t$ when $s_t < r$.

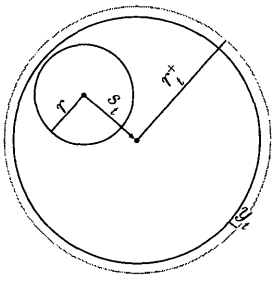
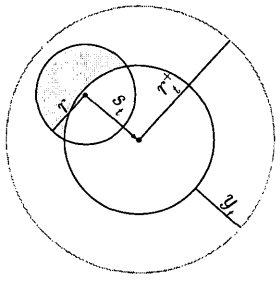
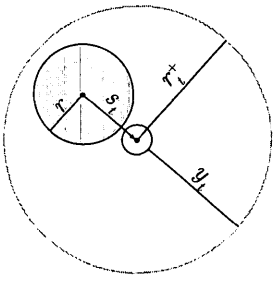
Intervals	$r_t^+ \geq r_t^+ - y_t \geq r + s_t$	$r + s_t > r_t^+ - y_t > s_t - r$	$s_t - r \geq r_t^+ - y_t \geq 0$
Region	$\emptyset$	$\mathcal{D}(p_t, r) \setminus \mathcal{I}(r, r_t^+ - y_t, s_t)$	$\mathcal{D}(p_t, r)$
Illustration			

Table 4.2: The intervals for $r_t^+ - y_t$ when $s_t \geq r$.

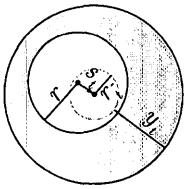
Intervals	$r_t^- \leq r_t^- + y_t \leq r - s_t$	$r - s_t < r_t^- + y_t < r + s_t$	$r + s_t \leq r_t^- + y_t < \infty$
Region	$\emptyset$	$\mathcal{D}(p_t, r) \setminus \mathcal{I}(r, r_t^- + y_t, s_t)$	$\mathcal{D}(p_0, r_t^- + y_t) \setminus \mathcal{D}(p_t, r)$
Illustration			

Table 4.3: The intervals for $r_t^- + y_t$ when $s_t < r$.

4.2.3 Conditional Expectation

Let X_0 be a Poisson process with intensity λ. The probability of k node positions occurring in a bounded set $\mathcal{X} \subset \mathbb{R}^2$ of size $\mu(\mathcal{X})$ is given in Equation 2.1.

Recall that δ is the distance a node moves between two consecutive time steps. Let the circle of radius r about point q be set of points $\mathcal{C}(q, r)$. The probability that a node positioned at q is mapped to a point in the bounded set $\mathcal{X} \subset \mathbb{R}^2$ is

$$k(q, \mathcal{X}) = \int_{\mathcal{C}(q,\delta)\cap\mathcal{X}} \frac{1}{2\pi\delta} dq', \tag{4.7}$$

The probability kernel $k(q, \mathcal{X})$ is consistent with the definition of isotropic node movements of fixed step length δ.

The mobility of the system of nodes positioned in $\mathbb{R}^2$ characterizes a discrete time stochastic point process $X = \{X_t : 0 \leq t \leq T\}$, where X_0 is a Poisson point process with intensity λ at time $t = 0$. This random sequence X is generated by repeated transformations, beginning with the initial point process X_0, where each point q is transitioned into a set $\mathcal{X}$ with probability $k(q, \mathcal{X})$ [55].

Lemma 1. *Let X_t be a Poisson process on $\mathbb{R}^2$ with intensity λ and suppose each point q of X_t is independently translated to a point in the bounded set $\mathcal{X} \subset \mathbb{R}^2$ with probability $k(q, \mathcal{X})$. The resulting point process X_{t+1} is a Poisson point process on $\mathbb{R}^2$ with intensity λ. Since the intensity λ for X_0 is an invariant measure of the*

probability kernel $k(q, \mathcal{X})$ the process X is a stationary Markov chain.

Proof. The result follows from Theorem 48 in Serfozo [55]. □

A stochastic process is **Markovian** if it is memoryless. This is also called the Markov property. By Lemma 1, X_t is a Poisson point process and the dynamics of X are Markovian.

If the random variable $X_t(\mathcal{E}_t(y_t|s_t))$ is the number of node positions in the annular region of width y_t at time t, then the event $X_t(\mathcal{E}_t(y_t|s_t)) = 0$ is the event of no node positions in the annular region at time t. The probability of this event is

$$\mathbb{P}[X_t(\mathcal{E}_t(y_t|s_t)) = 0|s_t] = e^{-\lambda \cdot \mu(\mathcal{E}_t(y_t|s_t))}. \tag{4.8}$$

We are interested in the event of at least one neighbour occurring in the annular region of width y_t. The set $\mathcal{E}_t(y_t|s_t))$ is a subset of the annular region and is the only place in the annular region where neighbour and non-neighbour nodes from time t may reside. The probability of this event is

$$\mathbb{P}[X_t(\mathcal{E}_t(y_t|s_t)) > 0|s_t] = 1 - \mathbb{P}[X_t(\mathcal{E}_t(y_t|s_t)) = 0|s_t] = 1 - e^{-\lambda \cdot \mu(\mathcal{E}_t(y_t|s_t))}. \tag{4.9}$$

Let random variable Y_t be the perturbation distance for the positive and negative positional constraints originating from time step t. The probability distribution for this variable is then

$$F_{Y_t}(y_t|s_t) = \mathbb{P}[Y_t \leq y_t|s_t] = 1 - e^{-\lambda \cdot \mu(\mathcal{E}_t(y_t|s_t))}. \tag{4.10}$$

The probability of at least one point in the region $\mathcal{E}_t(y_t|s_t)$ defines the probability distribution of Y_t, the perturbation distance before violating at least one positional constraint at time t. Potentially there are constraints from each time step and the bound requires that none of these constraints are violated by a common perturbation distance z.

Using Lemma 1, where X_t is Poisson and the process X is Markovian, an expression for the distribution of Z_T follows.

Theorem 1. *Given a sequence of resultants $s_1 \dots s_T$ after T time steps, the distribution for the perturbation distance $Z_T = \min\{Y_0 \dots Y_T\}$ is*

$$F_{Z_T}(z|s_1 \dots s_T) = 1 - e^{-\lambda\mu(\mathcal{E}_0(z))} \cdot \prod_{t=1}^{T} e^{-\lambda \int_{\mathcal{E}^c_{t-1}(z|s_{t-1})} k(q, \mathcal{E}^c_t(z|s_t))\mu(dq)}. \tag{4.11}$$

Proof. If random variable Z_T is $\min\{Y_0, \dots, Y_T\}$ then it has the following probability distribution,

$$F_{Z_T}(z|s_1 \dots s_T) =$$

$$\mathbb{P}[Z_T \leq z|s_1 \dots s_T] =$$

$$1 - \mathbb{P}[X_0(\mathcal{E}_t(z|s_0)) = 0, \dots, X_T(\mathcal{E}_T(z|s_T)) = 0] \quad .$$

By successive conditioning and the Markov property from Lemma 1

$$\mathbb{P}[X_0(\mathcal{E}_t(z|s_0)) = 0, \dots, X_T(\mathcal{E}_T(z|s_T)) = 0] =$$

$$\mathbb{P}[X_0(\mathcal{E}_t(z|s_0)) = 0] \cdot \prod_{t=1}^{T} \mathbb{P}[X_t(\mathcal{E}_t(z|s_t)) = 0|X_{t-1}(\mathcal{E}_{t-1}(z|s_{t-1})) = 0] \quad .$$

Now the event

$$[X_t(\mathcal{E}_t(z|s_t)) = 0|X_{t-1}(\mathcal{E}_{t-1}(z|s_{t-1})) = 0] \tag{4.12}$$

occurs when the region $\mathcal{E}_t(z|s_t)$ contain no nodes, given the region $\mathcal{E}_{t-1}(z|s_{t-1})$ contains no nodes. The probability of this event

$$\mathbb{P}[X_t(\mathcal{E}_t(z|s_t)) = 0 | X_{t-1}(\mathcal{E}_{t-1}(z|s_{t-1})) = 0] = e^{-\lambda \int_{\mathcal{E}^c_{t-1}(z|s_{t-1})} k(q, \mathcal{E}^c_t(z|s_t))\mu(dq)} \quad (4.13)$$

follows from the stationarity of the Poisson process with intensity λ. This is the probability that a typical point in the region $\mathcal{E}^c_{t-1}(z|s_{t-1})$ is not transitioned into the region $\mathcal{E}_t(z|s_t)$. This establishes the result. □

Further details of transformations of Poisson point process and the techniques of Theorem 1 are given in Theorem 48 and its subsequent discussion in Serfozo [55].

The expected perturbation distance Z_T conditional on the resultants $s_1 \dots s_T$ is

$$\mathbb{E}[Z_T | s_1 \dots s_T] = \int_0^{r_T^+} z \cdot f_{Z_T}(z|s_1 \dots s_T) dz. \quad (4.14)$$

4.2.4 The Case of Stationary Nodes

When nodes are stationary the embedding remains unchanged over time since $\delta = 0$ and all historical positional information becomes redundant. This case was illustrated in the beginning of Section 4.1. With no past information we have

$$F_{Z_0}(z) = \mathbb{P}[Z_0 \leq z] = 1 - \mathbb{P}[X_0(\mathcal{E}_0(z)) = 0] = 1 - e^{-\lambda \mu(\mathcal{E}_0(z))} = 1 - e^{-\lambda 4\pi r z} \quad (4.15)$$

and the expectation is then

$$\mathbb{E}[Z_0] = \int_0^a z \cdot F'_{Z_0}(z)dz \tag{4.16}$$

where $r_0^+ = r$. When the communication region are scaled to unit size, the particularly pleasing expression for the expected perturbation distance,

$$\mathbb{E}[Z_0] = \frac{1}{4\pi\lambda} - \frac{e^{-4\pi\lambda}}{1 - e^{-4\pi\lambda}} \tag{4.17}$$

is obtained for the bound on positional error [35].

4.2.5 Unconditional Expectation

We have initially conditioned on the resultant s_t which is the Euclidean distance the node has travelled between time step t and time step 0. The closer the position of node u at time t is to position of u at time step 0, the less impact a positional constraint from t will have on the perturbation distance. The probability distribution over these distances is a critical component of the analysis and will be developed next.

The previous section formulated a lower bound on the expected positional error, conditional on the sequence of resultants $\{s_1, \ldots, s_t\}$ between node u's current position $p_0(u)$ at time step $t = 0$ and its position $p_t(u)$ at time step t. The distance

s_t is a critical part of the bound. To see why, recall that the current position $p_0(u)$ is within a distance r_t^+ of the position of node u's neighbours at time t and not within a distance r_t^- of the positions of its non-neighbours. If the distance s_t between point $p_0(u)$ and point $p_t(u)$ is small then the positional constraints could be quite slack, especially for large t and δ. The positional constraint is slackened by a small distance s_t in the sense that it requires a greater perturbation distance to violate the constraint. On the other hand, a distance s_t that is close to $t\delta$ is a tenser (less slack) positional constraint since a relatively smaller perturbation distance is needed for a constraint violation.

Consider the movement model from Chapter 2, where during each time step a node alters its position by making an isotropic movement of a fixed length δ. The model is referred to as a random walk in a two-dimensional Euclidean space [16, 22]. The distance s_t from the originating position is the resultant after t movements and the change in distance $s_{t+1} - s_t$ between time step t and $t+1$ is called the **displacement**. The movement model implies the resultant at time step $t+1$ is dependent on the resultant from time step t. Figure 4.7 illustrates three movements of a random walk.

Let random variable S_t be the resultant after t steps. The resultant is determined by movements of a fixed step length and uniformly random direction in each time step. Consequently the probability distribution for the distance from the current

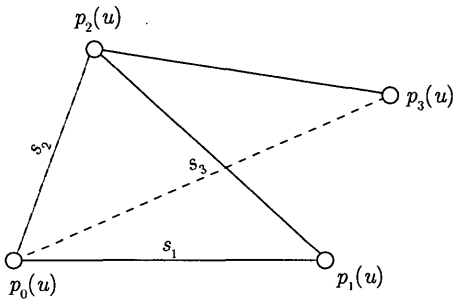


Figure 4.7: Random Walk: distance from the current position $p_0(u)$ of node u after three steps of length δ.

position depends on a single random variable, the direction of movement. The random walk problem has been well studied and we give the probability distribution for S_t below.

Lemma 2 (Dutka, [16]). *The probability distribution for the resultant S_t of a random walk after t steps of fixed length δ is given by*

$$\mathbb{P}[S_t \leq s] = F_{S_t}(s) = s \int_0^\infty J_1(su) J_0(\delta u)^t du, \tag{4.18}$$

and the density function is then

$$F'_{S_t}(s) = f_{S_t}(s) = s \int_0^\infty u J_0(su) J_0(\delta u)^t du, \tag{4.19}$$

where J_0 and J_1 are Bessel functions of the first kind.

After taking the marginal expectation over the random variables $S_1, \ldots, S_T$, the

unconditional expectation of Z for a fixed T is given by Theorem 2.

Theorem 2. *The expected perturbation distance after a sequence of T steps of fixed length δ under the random walk model is*

$$\mathbb{E}[Z] = \int_0^{T\delta} f_{S_T}(s_T) \cdot \ldots \left(\int_0^{2\delta} f_{S_2}(s_2) \cdot \left(\int_0^{1\delta} f_{S_1}(s_1) \cdot \mathbb{E}(Z|s_1 \ldots s_T) ds_1 \right) ds_2 \right) \ldots ds_T, \tag{4.20}$$

where $\mathbb{E}(Z|s_1 \ldots s_T)$ is given in Equation 4.14.

Corollary 1. *The expected perturbation distance given in Equation 4.20 is a lower bound on the expected positional error incurred by any connectivity-based positioning algorithm after a sequence of T isotropic steps of fixed length δ.*

Theorem 3 shows that this result also holds for a random walk with random step lengths uniformly distributed on the interval $[0, \delta]$, a more general model. There are two types of random walk to be considered: the random walk with fixed step length δ and the random walk with random step length Δ, where Δ has support in the range $[0, \delta]$. In both cases, each step is taken in a uniformly random direction. Theorem 3 uses Lemma 3 and Lemma 4, given below. Figure 4.8 illustrates some of the concepts in these results.

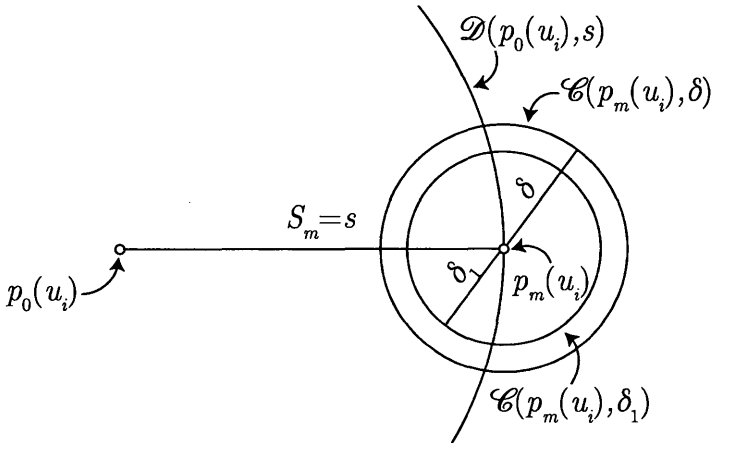


Figure 4.8: The resultant s after m steps of a random walk. If the random walker takes steps of fixed length δ, then its position after the $(m+1)^{\text{st}}$ step will be located on the circle $\mathcal{C}(p_m(u_i),\delta)$. If the walker takes fixed steps of length $\delta_1 \leq \delta$, then its position after the $(m+1)^{\text{st}}$ step will be located on the circle $\mathcal{C}(p_m(u_i),\delta_1)$. The resultant after the $(m+1)^{\text{st}}$ step will be larger than s if the walker's position is located outside the disk $\mathcal{D}(p_0(u_i),s)$.

Lemma 3. *Let the random variable S_{m+1} be the resultant after taking $m+1$ steps of length δ and the random variable S^*_{m+1} be the resultant after taking $m+1$ steps of length $0 < \delta_1 < \delta$. If $S_m = S^*_m = s > 0$, then $\mathbb{E}[S^*_{m+1}|S^*_m = s] \leq \mathbb{E}[S_{m+1}|S_m = s]$.*

Proof. Let the random variables $\theta^*_{m+1} \sim \text{Uniform}(0, 2\pi)$ and $\theta_{m+1} \sim \text{Uniform}(0, 2\pi)$ be the angles of the $(m+1)^{\text{st}}$ step of the random walk with step lengths δ_1 and the random walk with step lengths δ, respectively. The angle θ of the $(m+1)^{\text{st}}$ step for both random walks is with respect to the resultant $S_m = s$ and, without loss of generality, take s to be along the x-axis.

The resultants $S^*_{m+1}|S^*_m = s$ and $S_{m+1}|S_m = s$ will be compared with respect to

the angle $\theta \in [0, 2\pi]$ of the $(m+1)^{\text{st}}$ step. The range of the angle θ is partitioned into two subsets:

$$\begin{aligned} &\theta \in [0, \theta_1) \cup (2\pi - \theta_1, 2\pi] \quad (S^*_{m+1} | S^*_m = s) > (S_{m+1} | S_m = s), \\ &\theta \in [\theta_1, 2\pi - \theta_1] \qquad\qquad (S^*_{m+1} | S^*_m = s) \leq (S_{m+1} | S_m = s). \end{aligned} \tag{4.21}$$

The angle θ_1 is where $S^*_{m+1} = S_{m+1}$, which can be solved using the law of cosines,

$$(S^*_{m+1} | S^*_m = s, \theta^*_{m+1} = \theta_1) = (S_{m+1} | S_m = s, \theta_{m+1} = \theta_1) \tag{4.22}$$

$$s^2 + \delta_1^2 - 2s\delta_1 \cos(\theta_1) = s^2 + \delta^2 - 2s\delta \cos(\theta_1) \tag{4.23}$$

$$\cos(\theta_1) = \frac{\delta^2 - \delta_1^2}{2s(\delta - \delta_1)} \tag{4.24}$$

$$\cos(\theta_1) = \frac{\delta + \delta_1}{2s}. \tag{4.25}$$

Using the restrictions on δ,δ_1, and s, $\cos(\theta_1) = \frac{\delta+\delta_1}{2s} > 0$. If $\cos(\theta_1) = \frac{\delta+\delta_1}{2s} \geq 1$, then there does not exist a θ_1 where $S^*_{m+1} = S_{m+1}$. In this case $S^*_{m+1} \leq S_{m+1}$ for all $\theta \in [0, 2\pi]$. If $\cos(\theta_1) = \frac{\delta+\delta_1}{2s} < 1$, then $\theta_1 = \arccos\left(\frac{\delta+\delta_1}{2s}\right)$.

Assume $\cos(\theta_1) = \frac{\delta+\delta_1}{2s} < 1$ and consider any angle $\theta \in [0, \theta_1) \cup (2\pi - \theta_1, 2\pi]$. When $\theta \in [0, \theta_1)$ the difference of the resultants is

$$(S^*_{m+1}|S^*_m = s, \theta^*_{m+1} = \theta) - (S_{m+1}|S_m = s, \theta_{m+1} = \theta) = \tag{4.26}$$

$$(s^2 + \delta_1^2 - 2s\delta_1\cos(\theta)) - (s^2 + \delta^2 - 2s\delta\cos(\theta)) = \tag{4.27}$$

$$\delta_1^2 - \delta^2 + 2s\delta\cos(\theta) - 2s\delta_1\cos(\theta) = \tag{4.28}$$

$$\delta_1^2 - \delta^2 + (\delta - \delta_1)2s\cos(\theta) = \tag{4.29}$$

$$(\delta - \delta_1)(2s\cos(\theta) - (\delta + \delta_1)). \tag{4.30}$$

There is an angle $\theta^c \in (\pi - \theta_1, \pi]$ that is given by $\theta^c = \pi - \theta$ where the difference in resultants is

$$(S^*_{m+1}|S^*_m = s, \theta^*_{m+1} = \theta^c) - (S_{m+1}|S_m = s, \theta_{m+1} = \theta^c) = \tag{4.31}$$

$$(s^2 + \delta_1^2 - 2s\delta_1\cos(\theta^c)) - (s^2 + \delta^2 - 2s\delta\cos(\theta^c)) = \tag{4.32}$$

$$(s^2 + \delta_1^2 - 2s\delta_1\cos(\pi - \theta)) - (s^2 + \delta^2 - 2s\delta\cos(\pi - \theta)) = \tag{4.33}$$

$$(s^2 + \delta_1^2 + 2s\delta_1\cos(\theta)) - (s^2 + \delta^2 + 2s\delta\cos(\theta)) = \tag{4.34}$$

$$\delta_1^2 - \delta^2 + 2s\delta_1\cos(\theta) - 2s\delta\cos(\theta) = \tag{4.35}$$

$$\delta_1^2 - \delta^2 + (\delta_1 - \delta)2s\cos(\theta) = \tag{4.36}$$

$$(\delta - \delta_1)(-2s\cos(\theta) - (\delta + \delta_1)). \tag{4.37}$$

Let $g(\theta) = (\delta_1^2 - \delta^2 + (\delta - \delta_1)2s\cos(\theta))$. Then the expected difference in the resultants is

$$\mathbb{E}[(S^*_{m+1}|S^*_m = s) - (S_{m+1}|S_m = s)] = \tag{4.38}$$

$$\frac{1}{2\pi}\int_0^{2\pi} g(\theta)d\theta = \tag{4.39}$$

$$\frac{1}{2\pi}\int_0^{\theta_1} g(\theta)d\theta + \frac{1}{2\pi}\int_{\theta_1}^{\pi-\theta_1} g(\theta)d\theta + \tag{4.40}$$

$$\frac{1}{2\pi}\int_{\pi-\theta_1}^{\pi} g(\theta)d\theta + \frac{1}{2\pi}\int_{\pi}^{\pi+\theta_1} g(\theta)d\theta + \tag{4.41}$$

$$\frac{1}{2\pi}\int_{\pi+\theta_1}^{2\pi-\theta_1} g(\theta)d\theta + \frac{1}{2\pi}\int_{2\pi-\theta_1}^{2\pi} g(\theta)d\theta. \tag{4.42}$$

$$\tag{4.43}$$

For each angle $\theta \in [0, \theta_1)$ there exists an angle $\theta^c = \pi - \theta \in (\pi - \theta_1, \pi]$. Since

$$[(\delta - \delta_1)(2s\cos(\theta) - (\delta + \delta_1))] + [(\delta - \delta_1)(-2s\cos(\theta) - (\delta + \delta_1))] < 0, \tag{4.44}$$

we have $\frac{1}{2\pi}\int_0^{\theta_1} g(\theta)d\theta + \frac{1}{2\pi}\int_{\pi-\theta_1}^{\pi} g(\theta)d\theta < 0$.

Similarly, for each $\theta \in (2\pi - \theta_1, 2\pi]$ there exists an angle $\theta^c = \pi - \theta \in (\pi, \pi + \theta_1]$ and we have $\frac{1}{2\pi}\int_{2\pi-\theta_1}^{2\pi} g(\theta)d\theta + \frac{1}{2\pi}\int_{\pi}^{\pi+\theta_1} g(\theta)d\theta < 0$.

Finally, for all $\theta \in [\theta_1, \pi - \theta_1] \cup [\pi + \theta_1, 2\pi - \theta_1]$, $\frac{1}{2\pi}\int_{\theta_1}^{\pi-\theta_1} g(\theta)d\theta < 0$ and $\frac{1}{2\pi}\int_{\pi+\theta_1}^{2\pi-\theta_1} g(\theta)d\theta < 0$.

Therefore $\mathbb{E}[(S^*_{m+1}|S^*_m = s) - (S_{m+1}|S_m = s)] < 0$ and, by linearity of expectation, $\mathbb{E}[S^*_{m+1}|S^*_m = s] \leq \mathbb{E}[S_{m+1}|S_m = s]$.

□

Lemma 4. *Let the random variable S_{m+1} be the resultant after taking $m+1$ steps of length δ. If $s_1 \leq s_2$, then $\mathbb{E}[S_{m+1}|S_m = s_1] \leq \mathbb{E}[S_{m+1}|S_m = s_2]$.*

Proof. Let the random variables $\theta_1 \sim \text{Uniform}(0, 2\pi)$ and $\theta_2 \sim \text{Uniform}(0, 2\pi)$ be the angles of the $(m+1)^{\text{st}}$ step of the random walk conditioned on $S_m = s_1$ and the random walk conditioned on $S_m = s_2$, respectively. Without loss of generality, let angle θ of the $(m+1)^{\text{st}}$ step for both random walks is with respect to the x-axis.

The resultants $S_{m+1}|S_m = s_1$ and $S_{m+1}|S_m = s_2$ will be compared with respect to the angle $\theta \in [0, 2\pi]$ of the $(m+1)^{\text{st}}$ step. If $s_1 = s_2$, then $\mathbb{E}[S_{m+1}|S_m = s_1] = \mathbb{E}[S_{m+1}|S_m = s_2]$.

Assuming $s_1 < s_2$, the angle θ' where $S_{m+1}|S_m = s_1 = S_{m+1}|S_m = s_2$ is

$$(S_{m+1}|S_m = s_1, \theta_1 = \theta') = (S_{m+1}|S_m = s_2, \theta_2 = \theta') \tag{4.45}$$

$$(s_1^2 + \delta^2 - 2s_1\delta\cos(\theta')) - (s_2^2 + \delta^2 - 2s_2\delta\cos(\theta')) \tag{4.46}$$

$$\cos(\theta') = \frac{s_2 + s_1}{2\delta}. \tag{4.47}$$

For any angle $\theta \in [0, 2\pi]$, the difference of the resultants is

$$(S_{m+1}|S_m = s_1, \theta_1 = \theta) - (S_{m+1}|S_m = s_2, \theta_2 = \theta) = \tag{4.48}$$

$$(s_1^2 + \delta^2 - 2s_1\delta\cos(\theta)) - (s_2^2 + \delta^2 - 2s_2\delta\cos(\theta)) = \tag{4.49}$$

$$s_1^2 - s_2^2 - 2s_1\delta_1\cos(\theta) + 2s_2\delta\cos(\theta) = \tag{4.50}$$

$$s_1^2 - s_2^2 + (s_2 - s_1)2\delta\cos(\theta) = \tag{4.51}$$

$$(s_1 - s_2)((s_1 + s_2) - 2\delta\cos(\theta)). \tag{4.52}$$

Let the difference in the two resultants be $g(\theta) = (s_1 - s_2)((s_1 + s_2) - 2\delta\cos(\theta))$. The resultants $(S_{m+1}|S_m = s_1, \theta_1 = \theta)$ and $(S_{m+1}|S_m = s_2, \theta_2 = \theta)$ will be compared with respect to the angle $\theta \in [0, 2\pi]$ of the $(m+1)^{\text{st}}$ step.

If $\cos(\theta') = \frac{s_2+s_1}{2\delta} > 1$ then there does not exist a θ' and in this case $(S_{m+1}|S_m = s_1, \theta_1 = \theta) < (S_{m+1}|S_m = s_2, \theta_2 = \theta)$ for all θ since the difference in the two resultants is $g(\theta) = (s_1 - s_2)((s_1 + s_2) - 2\delta\cos(\theta)) < 0$. Therefore, in this case, $\mathbb{E}[(S_{m+1}|S_m = s_1) - (S_{m+1}|S_m = s_2)] < 0$ and, by linearity of expectation, $\mathbb{E}[S_{m+1}|S_m = s_1] \leq \mathbb{E}[S_{m+1}|S_m = s_2]$.

Assume $\cos(\theta') = \frac{s_2+s_1}{2\delta} \leq 1$ and partition the range of the angle θ into two subsets:

$$\theta \in [0, \theta') \cup (2\pi - \theta', 2\pi] \quad (S_{m+1}|S_m = s_1, \theta_1 = \theta) > (S_{m+1}|S_m = s_2, \theta_2 = \theta),$$

$$\theta \in [\theta', 2\pi - \theta'] \qquad (S_{m+1}|S_m = s_1, \theta_1 = \theta) \leq (S_{m+1}|S_m = s_2, \theta_2 = \theta). \tag{4.53}$$

The expected difference in the resultants is

$$\mathbb{E}[(S_{m+1}|S_m = s_1) - (S_{m+1}|S_m = s_2)] = \tag{4.54}$$

$$\frac{1}{2\pi}\int_0^{2\pi} g(\theta)d\theta = \tag{4.55}$$

$$\frac{1}{2\pi}\int_0^{\theta'} g(\theta)d\theta + \frac{1}{2\pi}\int_{\theta'}^{\pi-\theta'} g(\theta)d\theta + \tag{4.56}$$

$$\frac{1}{2\pi}\int_{\pi-\theta'}^{\pi} g(\theta)d\theta + \frac{1}{2\pi}\int_{\pi}^{\pi+\theta'} g(\theta)d\theta + \tag{4.57}$$

$$\frac{1}{2\pi}\int_{\pi+\theta'}^{2\pi-\theta'} g(\theta)d\theta + \frac{1}{2\pi}\int_{2\pi-\theta'}^{2\pi} g(\theta)d\theta. \tag{4.58}$$

$$\tag{4.59}$$

Consider any angle $\theta \in [0, \theta') \cup (2\pi - \theta', 2\pi]$.

For each $\theta \in [0, \theta')$, $g(\theta) = (s_1 - s_2)((s_1 + s_2) - 2\delta\cos(\theta))$. There is a complementary angle $\theta^c \in (\pi - \theta', \pi]$ where $g(\theta^c) = (s_1 - s_2)((s_1 + s_2) + 2\delta\cos(\theta))$.

Since

$$[(s_1 - s_2)((s_1 + s_2) - 2\delta\cos(\theta))] + [(s_1 - s_2)((s_1 + s_2) + 2\delta\cos(\theta))] < 0, \quad (4.60)$$

we have $\frac{1}{2\pi}\int_0^{\theta'} g(\theta)d\theta + \frac{1}{2\pi}\int_{\pi-\theta'}^{\pi} g(\theta)d\theta < 0$.

Similarly, for each $\theta \in (2\pi - \theta', 2\pi]$ there exists a complementary angle $\theta^c = \pi - \theta \in (\pi, \pi + \theta']$ and we have $\frac{1}{2\pi}\int_{2\pi-\theta'}^{2\pi} g(\theta)d\theta + \frac{1}{2\pi}\int_{\pi}^{\pi+\theta'} g(\theta)d\theta < 0$.

Finally, for all $\theta \in [\theta', \pi - \theta'] \cup [\pi + \theta', 2\pi - \theta']$, $\frac{1}{2\pi}\int_{\theta'}^{\pi-\theta'} g(\theta)d\theta < 0$ and $\frac{1}{2\pi}\int_{\pi+\theta'}^{2\pi-\theta'} g(\theta)d\theta < 0$.

Therefore $\mathbb{E}[(S_{m+1}|S_m = s_1) - (S_{m+1}|S_m = s_2)] < 0$ and, by linearity of expectation, $\mathbb{E}[S_{m+1}|S_m = s_1] \leq \mathbb{E}[S_{m+1}|S_m = s_2]$. □

Theorem 3. *Let the random variable S_m be the resultant after taking m steps of length δ and the random variable S_m^* be the resultant after taking m steps of length $0 < \delta_1 \leq \delta$. $\mathbb{E}[S_m^*] \leq \mathbb{E}[S_m]$.*

Proof. The resultant after $m+1$ steps can is a random variable $S_{m+1} = S_m + X_{m+1}$. The conditional distribution for $S_{m+1}|S_m = s_m$ can be written

$$\mathbb{P}[S_{m+1} \leq s_{m+1} | S_m = s_m] = \tag{4.61}$$

$$\mathbb{P}[S_m + X_{m+1} \leq s_{m+1} | S_m = s_m] = \tag{4.62}$$

$$\mathbb{P}[s_m + X_{m+1} \leq s_{m+1} | S_m = s_m] = \tag{4.63}$$

$$\mathbb{P}[X_{m+1} \leq (s_{m+1} - s_m) | S_m = s_m]. \tag{4.64}$$

The conditional expectation for $S_{m+1}|S_m = s_m$ can be written

$$\mathbb{E}[S_{m+1} | S_m = s_m] = \tag{4.65}$$

$$\int_0^\infty s_{m+1} \cdot f_{S_{m+1}}(s_{m+1} | S_m = s_m) ds_{m+1} = \tag{4.66}$$

$$\int_0^\infty s_{m+1} \cdot f_{X_{m+1}}((s_{m+1} - s_m) | S_m = s_m) ds_{m+1}. \tag{4.67}$$

Using the conditional expectation, the expected value of S_{m+1} is

$$\mathbb{E}[S_{m+1}] =$$

$$\int_0^\infty f_{S_m}(s_m) \cdot \mathbb{E}[S_{m+1}|S_m = s_m] ds_m =$$

$$\int_0^\infty f_{S_m}(s_m) \cdot \left(\int_0^\infty s_{m+1} \cdot f_{X_{m+1}}((s_{m+1} - s_m)|S_m = s_m) ds_{m+1} \right) ds_m.$$

The argument is completed by induction on m. $\mathbb{E}[S_1^*] \leq \mathbb{E}[S_1]$ by definition. Let's assume $\mathbb{E}[S_m^*] \leq \mathbb{E}[S_m]$. Recall $\mathbb{E}[S_m] = \int_0^\infty s \cdot f_{S_m}(s)$. Now by the induction hypothesis

$$\int_0^\infty s \cdot f_{S_m^*}(s) \leq \int_0^\infty s \cdot f_{S_m}(s) \tag{4.68}$$

and by Lemma 4, $\mathbb{E}[S_{m+1}|S_m = s_1] \leq \mathbb{E}[S_{m+1}|S_m = s_2]$ where $s_1 \leq s_2$, so

$$\mathbb{E}[S_{m+1}^*] = \int_0^\infty \mathbb{E}[S_{m+1}^*|S_m^* = s] \cdot f_{S_m^*}(s) ds \leq \int_0^\infty \mathbb{E}[S_{m+1}^*|S_m^* = s] \cdot f_{S_m}(s) ds. \tag{4.69}$$

By Lemma 3, $\mathbb{E}[S_{m+1}^*|S_m^* = s] \leq \mathbb{E}[S_{m+1}|S_m = s]$, so

$$\int_0^\infty \mathbb{E}[S^*_{m+1}|S^*_m = s] \cdot f_{S_m}(s)ds \leq \int_0^\infty \mathbb{E}[S_{m+1}|S_m = s] \cdot f_{S_m}(s)ds = \mathbb{E}[S_{m+1}], \tag{4.70}$$

which establishes the claim. □

Theorem 3 compares a random walk taking fixed step lengths δ_1 with a random walk taking fixed step length δ. Suppose the step length for the resultant S^*_t is a random variable Δ with support in the range $[0, \delta]$, instead of a fixed step length $0 < \delta_1 \leq \delta$, then Theorem 1 still holds. If f_Δ is the probability density function of the random variable Δ, then we have $\mathbb{E}[S^*_t] = \int_0^\delta f_\Delta(\delta_1) \cdot \mathbb{E}[S^*_t|\Delta = \delta_1]d\delta_1 \leq \int_0^\delta f_\Delta(\delta_1) \cdot \mathbb{E}[S_t|\Delta = \delta]d\delta_1 = \mathbb{E}[S_t|\Delta = \delta] \cdot \int_0^\delta f_\Delta(\delta_1)d\delta_1 = \mathbb{E}[S_t|\Delta = \delta] = \mathbb{E}[S_t]$.

Though the random walk model defined in Section 2 uses fixed step lengths δ, Theorem 3 implies that the result also applies to a random walk with random step lengths on the interval $[0, \delta]$.

Comparing the expected distance $\mathbb{E}[S_t]$ between $p_t(u_i)$ and $p_0(u_i)$ with the maximum distance $t\delta$ reveals a growing gap between the two as t increases. The maximum distance is used to impose constraints on the position of a node. This gap suggests that a larger perturbation is needed, in expectation, to violate a constraint (see Figure 4.4b). Since Z_T is the minimum perturbation distance, this suggests there is a limit to the usefulness of past positional information. The effect of past

positional information on the perturbation distance is analyzed in Section 4.4.

4.3 Analysis of the Positional Error

The expression for the expected perturbation distance $\mathbb{E}[Z_T]$ in Equation 4.20 depends on the model parameters: the intensity λ, step length δ, and the number of time steps T. This equation provides a mechanism for evaluating the impact of model parameters on the bound on positional error. The equation is not amenable to symbolic manipulation, therefore we proceed with a numeric evaluation of Equation 4.20 for various settings of λ, δ, and T.

The use of negative positional constraints in practice has previously been questioned [15, 4]. The central criticism follows from the fact that the absence of received messages between two nodes does not imply they are not within a distance r of one another. The two most commonly offered examples are environmental obstacles that interfere with radio communication and anisotropic antenna configurations. These situations can incorrectly turn a positive positional constraint into a negative positional constraint. Atmospheric conditions can also affect radio signal propagation. Stationary node settings offer the possibility of correcting misidentified positional constraints due to atmospheric conditions by repeated transmissions. However, systems of mobile nodes make potential correction of misidentified positional constraints less likely since the communication links are constantly changing.

In this evaluation of model parameters we do not make use of negative positional constraints.

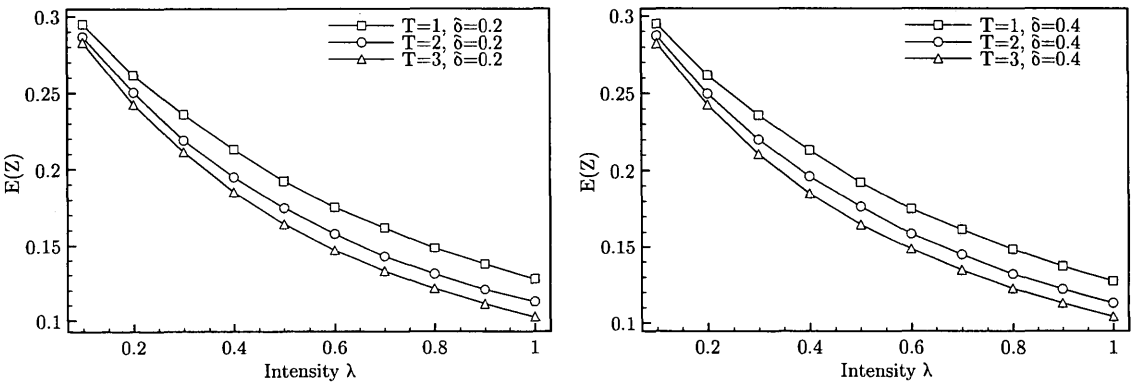
If a system is designed where negative constraints are removed from consideration the bound can be adapted by changing the region devoid of node positions from $\mathcal{E}_t(y_t|s_t) = \mathcal{E}_t^+(y_t|s_t) \cup \mathcal{E}_t^-(y_t|s_t)$ to the region $\mathcal{E}_t^+(y_t|s_t)$ in the probability distribution given in Equation 4.11.

Key components in Equation 4.11 are the size of regions $\mathcal{E}_t^+(z)$ and $\mathcal{E}_{t-1}^+(z)$ and the set operations on those regions. Equation 4.20 has been evaluated using numerical techniques to find the expected perturbation distance $\mathbb{E}[Z_T]$.

Figure 4.9 plots the minimum expected positional error of a connectivity-based positioning algorithm for T steps of positioning history using positive positional constraints. Since we are comparing the effect of added historical positional information, the base case used for comparison where there is no positional history requires the existence of a neighbour at time step $t = 0$. Therefore we condition on the existence of at least one neighbour in the most recent time step

$$F_{Z_T}(z|X_0(\mathcal{E}_0^+(r)) > 0), \tag{4.71}$$

which allows for comparison with the minimum expected positional error when there is no positional history.



(a) Step Length $\delta = 0.2$.

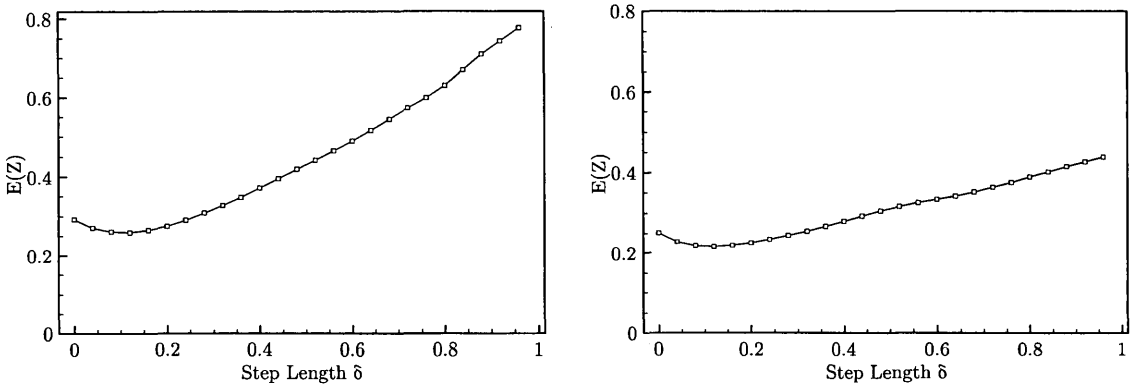
(b) Step Length $\delta = 0.4$.

Figure 4.9: The value of historical information for various step lengths δ.

The smaller the step length δ, the less slack there is in the constraints from past time steps. The discussion from the end of Section 4.2.5 adds further insight to this observation.

The use of past positional information beyond a few time steps is of little benefit in reducing the expected perturbation distance. This is primarily a consequence of the slackening of positional constraints due to movement and is analyzed in greater detail in the Section 4.4.

Figure 4.10 gives a better illustration of the impact small step lengths have on the expected perturbation distance. Initially, a decrease in $\mathbb{E}[Z]$ occurs with small step lengths but as the step size grows the expected perturbation distance eventually does as well. The J-curve shape in the graph that appears within a small range of step lengths is an empirical observation previously made of positioning algorithms



(a) Intensity $\lambda = 0.2$.

(b) Intensity $\lambda = 0.4$.

Figure 4.10: The impact of node speed on the expected perturbation distance for $T = 3$.

in mobile systems [4, 64, 53, 21, 66, 40]. The larger step sizes add more slack to past positional constraints making them less beneficial for reducing positional error.

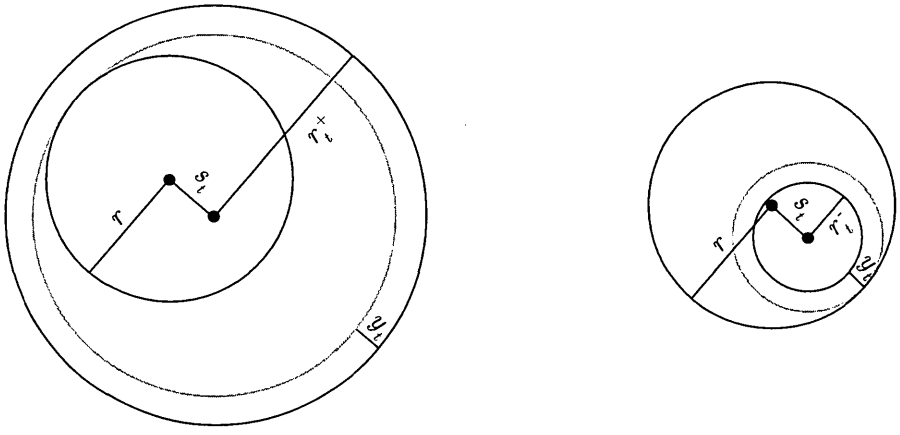
4.4 Number of Useful Time Steps

The added value of historical positioning information is of particular interest to the performance of a positioning algorithm. This section analyzes the effectiveness of past positional information in reducing the perturbation distance.

Let random variable Y_t be the perturbation distance (using both positive and negative positional information) for time step t and $Z_T = \min\{Y_0, Y_1, ..., Y_T\}$. The distance between the current position $p_0(u_i)$ of node u_i and its position $p_t(u_i)$ at time t is the random variable S_t.

The constraints from time step t that are imposed on the current position of

a node u_i restrict the maximum (or minimum) distance between the node and its former neighbours $N_t(u_i)$ (or former non-neighbours $N_t^c(u_i)$) to be $r^+ = r + t\delta$ (or $r^- = r - t\delta$). The **gap** is the difference between maximum distance between $p_t(u_i)$ and $p_0(u_i)$, and the actual distance S_t. Using the definition of positive constraints, the gap is $r_t^+ - (S_t + r) = r + t\delta - S_t - r = t\delta - S_t$. The gap is equivalent when using negative constraints. The gap is a random variables because it is defined using the random variable S_t.



(a) A gap of size y_t at time step t using positive positional information.

(b) A gap of size y_t at time step t using negative positional information.

Figure 4.11: An illustration of a gap of size y_t at time step t when $S_t = s_t$.

Figure 4.11 illustrates the gap for resultant $S_t = s_t$. The perturbation distance Y_t for this time step is in the interval $[0, r+]$. The perturbation distance Y_t for time step t is the maximum distance node u_i could be moved without violating

a constraint. If $S_t = s_t$, then the inner annular region of width $t\delta - s_t$ does not intersect the disk shaped communication region $\mathcal{D}(p_t(u_i), r)$, where the set of neighbour nodes $N_t(u_t)$ must reside, (and the outer annular region does not intersect the region $\mathcal{D}^c(p_t(u_i), r)$) from time t. Therefore $\mathbb{P}[Y_t > t\delta - s_t] = 1$ because the annular region cannot contain any nodes.

The gap is used to analyze the impact of constraints from additional time steps on the perturbation distance. For the sequence $\{Y_0, \ldots, Y_T\}$ let $Z_t = \min\{Y_0, \ldots, Y_t\}$, the minimum of the first $t+1$ elements of the sequence. The random variable

$$W_t = ((t+1)\delta - S_{t+1}) - Z_t \tag{4.72}$$

is the difference between the gap at $t+1$ and the minimum perturbation distance Z_t up to time step t. We have the following stochastic ordering result on W_t.

Theorem 4. $\mathbb{P}[W_t \leq w] \leq \mathbb{P}[W_{t-1} \leq w]$ *for each* w.

Proof. Let us consider $W_t - W_{t-1}$, then from Equation 4.72 we can write

$$W_t = W_{t-1} + (S_t - S_{t+1} + \delta) + (Z_{t-1} - Z_t). \tag{4.73}$$

Since the minimum of $\{Y_0, \ldots, Y_{t-1}\}$ can only decrease with inclusion of more time steps from the sequence we obtain the following inequality for each w

$$\mathbb{P}[W_t \leq w] \leq \mathbb{P}[W_{t-1} + (S_t - S_{t+1} + \delta) \leq w]. \tag{4.74}$$

The smallest displacement between consecutive time steps t and $t+1$ is $-\delta$ so

$$\mathbb{P}[W_t \leq w] \leq \mathbb{P}[W_{t-1} \leq w] \tag{4.75}$$

which establishes the claim. □

Theorem 4 implies that the difference between the gap and perturbation distance can only grow in expectation with the number of time steps t. The event $W_t \geq 0$ occurs when the perturbation distance Y_t for time step t does not exceed the gap at t. The probability of this event is the probability that the minimum perturbation distance is one of $Y_0, \ldots, Y_t$. If this probability is high, then there is a low probability that constraints from time steps past t will reduce the perturbation distance.

The random variables Z_t and S_{t+1} are conditionally independent given $S_1, \ldots, S_t$. The conditional probability distribution for random variable W_t can be described by the convolution of their respective conditional distributions,

$$F_{W_i}(w|s_1, \ldots, s_t) = \int_0^{r_t^+} \left[\int_{s_t+\delta-z-w}^{s_t+\delta} f_{S_{t+1}}(s_{t+1}|s_t) ds_{t+1} \right] f_{Z_t}(z|s_1, \ldots, s_t) dz. \tag{4.76}$$

Consider initially the random variable W_1: the difference between the perturbation distance up to time one and the gap at time two. The distribution of the resultant S_2 after two steps of length δ, which is a component of W_1, is given by Proposition 1.

Proposition 1 (Hughes, [22]). *The probability distribution for the resultant S_2 after two steps of fixed length δ is*

$$F_{S_2}(s) = \frac{2}{\pi}\arcsin(\frac{s}{2\delta}) \tag{4.77}$$

and the density function for the resultant S_2 is then,

$$F'_{S_2}(s) = f_{S_2}(s) = \frac{2}{\pi}\frac{1}{\sqrt{4\delta^2 - s^2}}, 0 < s < 2\delta. \tag{4.78}$$

Since the resultant S_1 takes a value of $s_1 = \delta$ with probability one, from Equation 4.11, the probability distribution of Z_1 is

$$F_{Z_1}(z) = F_{Z_1}(z|s_1) \tag{4.79}$$

The distribution for W_1 is given by the convolution of Z_1 and S_2,

$$F_{W_1}(w) = \int_0^{\delta+r}\left[\int_{s_1+\delta-z-w}^{s_1+\delta} f_{S_2}(s)ds\right] f_{Z_1}(z)dz \tag{4.80}$$

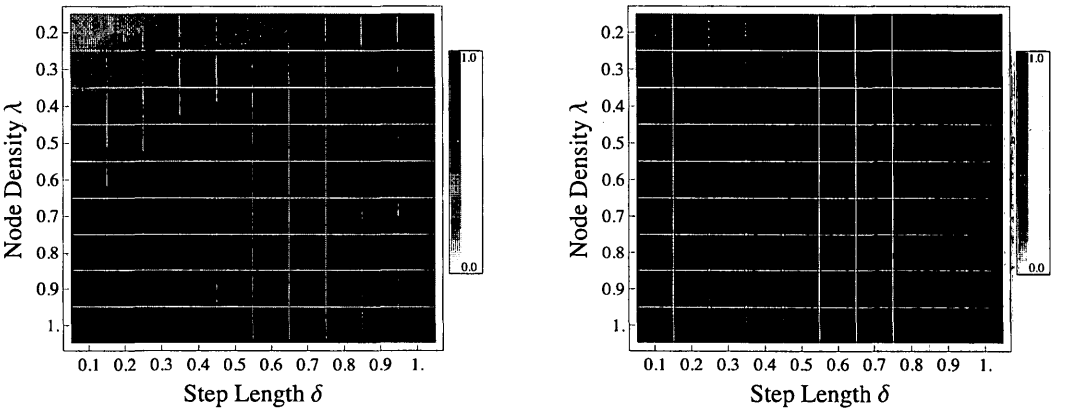
Equation 4.80 is a distribution that can be evaluated at various values of λ and δ to calculate the probability of the event $W_1 \geq 0$. Table 4.4 contains the probabilities for this event while Table 4.5 shows the probabilities for the event $W_2 \geq 0$.

		Node Density λ									
		0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
Fixed Step Length δ	0.20	0.31	0.34	0.36	0.37	0.39	0.43	0.45	0.48	0.50	0.52
	0.30	0.35	0.43	0.46	0.48	0.49	0.50	0.53	0.55	0.57	0.61
	0.40	0.46	0.47	0.54	0.55	0.56	0.57	0.59	0.62	0.64	0.65
	0.50	0.49	0.53	0.56	0.58	0.61	0.63	0.64	0.65	0.67	0.68
	0.60	0.52	0.54	0.58	0.61	0.62	0.64	0.66	0.68	0.71	0.73
	0.70	0.54	0.55	0.61	0.62	0.63	0.67	0.68	0.73	0.72	0.76
	0.80	0.56	0.58	0.62	0.64	0.67	0.70	0.71	0.72	0.73	0.76
	0.90	0.57	0.59	0.63	0.65	0.68	0.71	0.72	0.74	0.75	0.77
	1.00	0.58	0.61	0.65	0.66	0.68	0.71	0.73	0.75	0.76	0.78

Table 4.4: The probability of $W_1 \geq 0$ using only positive positional constraints for various values of λ and δ.

		Node Density λ									
		0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
Fixed Step Length δ	0.20	0.54	0.57	0.61	0.66	0.69	0.71	0.73	0.75	0.78	0.80
	0.30	0.63	0.67	0.72	0.74	0.77	0.79	0.81	0.83	0.84	0.86
	0.40	0.72	0.75	0.78	0.81	0.82	0.85	0.86	0.88	0.90	0.92
	0.50	0.73	0.75	0.79	0.82	0.84	0.86	0.88	0.89	0.90	0.92
	0.60	0.75	0.79	0.81	0.83	0.86	0.89	0.90	0.91	0.92	0.92
	0.70	0.77	0.80	0.82	0.83	0.86	0.90	0.91	0.92	0.92	0.94
	0.80	0.77	0.80	0.84	0.85	0.88	0.91	0.92	0.93	0.94	0.95
	0.90	0.78	0.81	0.84	0.87	0.89	0.92	0.93	0.94	0.95	0.96
	1.00	0.80	0.82	0.85	0.89	0.90	0.91	0.93	0.94	0.95	0.96

Table 4.5: The probability of $W_2 \geq 0$ using only positive positional constraints for various values of λ and δ.



(a) $\mathbb{P}[W_1 \geq 0]$ (b) $\mathbb{P}[W_2 \geq 0]$

Figure 4.12: An illustration of the probability of $W_1 \geq 0$ and the probability of $W_2 \geq 0$ for various values of node density λ and step length δ. The darker the cells the greater the probability of the event.

Examining Table 4.4 it can be seen that $\mathbb{P}[W_1 \geq 0]$ increases with increasing node density λ and node speed δ and that for the majority of parameter values, $\mathbb{P}[W_1 \geq 0] > 0.5$. For smaller values of λ and δ the expected perturbation distance may be reduced with additional time steps. However, as shown in Table 4.5, the probability that positional information is beneficial beyond three time steps is even less likely. Figure 4.12 illustrates the probability of each event. If positional information is marked with the time where it was acquired, the algorithm designer could find this information of practical value since it suggests a limit on the usefulness of past positional information.

4.5 Discussion

In addition to bounding the minimum expected positional error of connectivity-based positioning algorithms in mobile settings, the analysis highlights some important characteristics of such a system that are valuable for algorithm design.

4.5.1 Observations from the Model

Some observations can be made of the connectivity and mobility model used in this analysis that impact the design of connectivity-based positioning algorithms. Recall that the model states that the connectivity graph is determined locally and therefore any positional constraints must be disseminated to other nodes by the exchange of messages. This implies that the determination of positional constraints must be performed in a distributed manner which has consequences for positioning algorithms. Some of these consequences are discussed in the following observations.

Observation 1 (Necessary Neighbour Condition). *All constraint information is discovered by direct communication with other nodes. Therefore, at least one neighbour is required to communicate with non-neighbours at a given step.*

If a node is isolated it cannot communicate with other nodes and therefore cannot learn of the existence of any non-neighbours.

Observation 2. *Negative constraints that do not affect the size of the region of uncertainty are redundant.*

Observation 1 states that at least one a neighbour is required in step t to obtain negative constraint information in step t. Therefore negative constraints that are greater than a distance $3r$ from the node position at step t will be redundant.

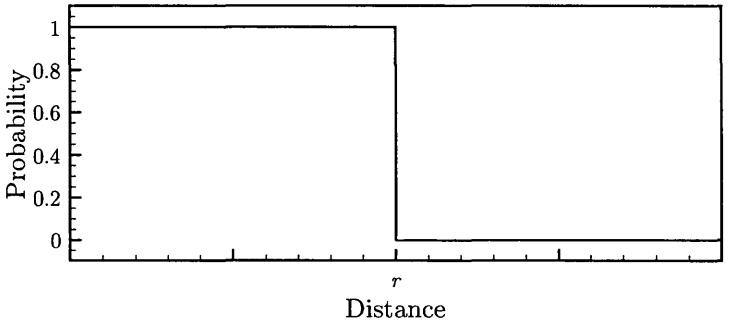
Recall that in Section 4.2 the upper limit on the integration was set to $a = r_t^+$. Observation 1 justifies the assignment $a = r_t^+$ as an upper limit on the size of the perturbation distance z in step t.

Chapter 5

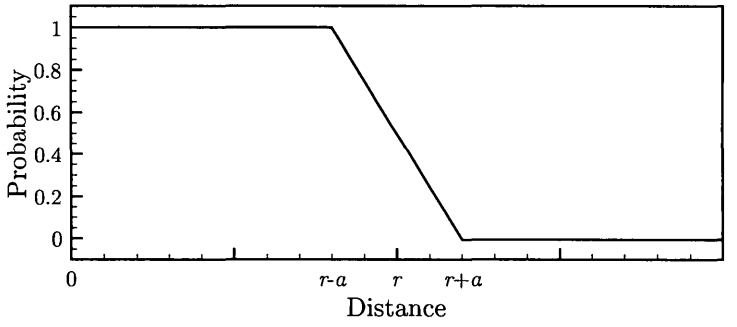
A Probabilistic Model for Node Communication

Chapter 4 used the Unit Disk model to analyze the positional error inherent to the positioning problem with connectivity-based constraints. Another communication model commonly used is Noisy Disk, a model which generalizes the Unit Disk model by introducing uncertainty to node connectivity. The probability distribution of both models is given in Figure 5.1.

Neither of these models have been based directly on the communication properties of wireless sensor devices, but rather indirectly on the assumed properties of radio hardware. An alternative model for node communication is presented in this chapter. The Empirical Disk is a probabilistic model for node communication that is generated from sensor connectivity observations. The impact of node connectivity on positioning under these three models is examined in Chapter 6.



(a) Probability distribution for the Unit Disk model.



(b) Probability distribution for the Noisy Disk model with irregularity parameter $\alpha = 0.20$.

Figure 5.1: The probability distributions for communication models with a communication range $r = 1$.

5.1 Sensor Connectivity

The connectivity between a pair of nodes at any distance is uncertain and is specific to a sensor device. A radio communication experiment was conducted to collect connectivity data for the purpose of developing a probabilistic model of connectivity. An experiment was performed for the sensors based on the Nordic and Texas Instruments hardware. Each experiment involved source node u_0 that sends mes-

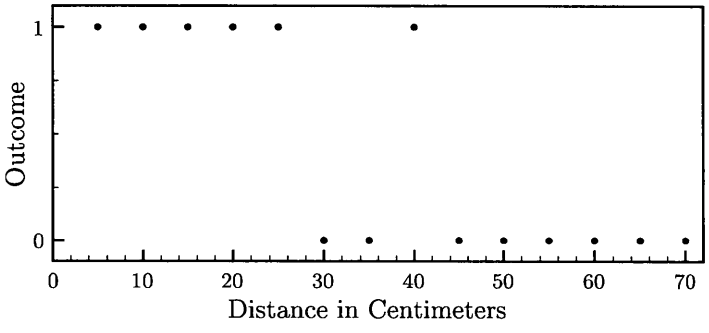
sages and a target node u_1 that may receive messages. The target node is placed at one of a set of fixed distances from the source node. The position of the target node and the set of distances define a set of concentric circles about the position of the source node with radii equal to each distance. The target node is placed at each one of a set of eight positions on each circle. The positions on each circle are evenly spaced. The position and orientation of source node u_0 is fixed. At each position for the target node, if at least one of the 5 packets transmitted by node u_0 is received by node u_1 then the nodes are deemed connected.

The plots in Figure 5.2 give the connectivity observations from two different trials of the radio communication experiment. A '1' was recorded if a message was received during a trial or a '0' if no message was received. The pattern of communication is consistent with the idea that nearby nodes are more likely to be able to communicate than distant nodes.

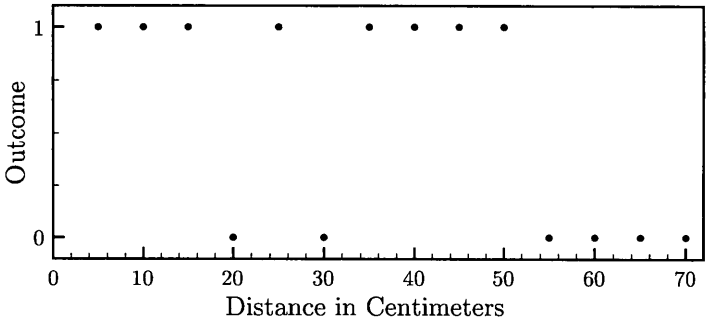
Let Y_{N} be a Bernoulli random variable that takes a value of 1 if the pair of nodes are connected and value 0 if the pair nodes are not connected, and x be the distance between the nodes u_0 and u_1. Then,

$$\pi_{\mathrm{N}}(x) = \mathbb{P}[Y_{\mathrm{N}} = 1]. \tag{5.1}$$

Logistic regression is a popular technique for modelling categorical data and is



(a) Scatter plot of successful message transmissions versus distance.



(b) Scatter plot of successful message transmissions versus distance.

Figure 5.2: The results from two different sender-receiver configurations during the radio communication experiment with the Nordic hardware.

used to model the connectivity data obtained from the radio experiments. Concordance is a standard measure of classification accuracy of a logistic model that describes the level of agreement between the model and data [52].

Highly predictive probabilistic models were developed for both sensor devices based on the connectivity data from the radio experiments. The position and orientation of the target node u_1 were not found to affect predictability.

The following logistic model is an excellent fit to the connectivity data for the Nordic hardware

$$\pi_{\mathrm{N}}(x) = \frac{e^{\alpha_0+\alpha_1 x}}{1+e^{\alpha_0+\alpha_1 x}}. \tag{5.2}$$

The estimated parameters α_0, α_1 for the $\mathrm{ED_N}$ communication model that is based on the Nordic hardware are given in Table 5.1 and a plot of the resulting function is given in Figure 5.3. The model is 93.9% concordant with the observed data which makes the model highly predictive. This communication model is used in Chapter 7 for comparing the performance of an algorithm executing on a sensor network and in a computer simulation. Two radio properties that affect the distribution are transmission power and radio sensitivity. The transmission power is an adjustable parameter of a radio that determines the output power to the antenna. The receiver sensitivity is the minimum strength required to successfully process a received signal. The threshold for successfully processing a signal is reflected in Figure 5.3 by the sudden transition from successful message transmission to unsuccessful message transmission as the distance between the sensor devices increases.

The following logistic model is an excellent fit to the connectivity data for the Texas Instruments hardware,

Parameter	Estimate	Standard Error
α_0	7.351×10^{0}	4.852×10^{-1}
α_1	-1.992×10^{-1}	1.284×10^{-2}

Table 5.1: Parameter estimates from a binary logistic regression on the observed connectivity data on the sensor devices based on the Nordic hardware.

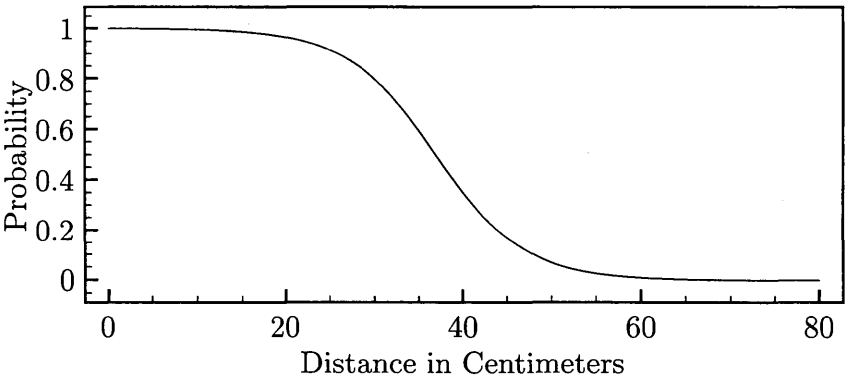


Figure 5.3: The probability of a connection between a pair of nodes in an obstacle free environment according to the empirical disk ($\mathrm{ED_N}$) model.

$$\pi_{\mathrm{TI}}(x) = \frac{e^{\beta_0+\beta_1 x^2}}{1+e^{\beta_0+\beta_1 x^2}}. \tag{5.3}$$

The estimated parameters β_0, β_1 for the $\mathrm{ED_{TI}}$ communication model that is based on the Texas Instruments hardware are given in Table 5.2. The model is 90.1% concordant with the data implying the model is highly predictive. This model is used for the computer simulation experiments described in Section 6.4. The distribution for this model is plotted in Figure 5.4 and it experiences the

transition in connectivity for the same reasons as the Nordic hardware.

Parameter	Estimate	Standard Error
β_0	6.556×10^{0}	3.057×10^{-1}
β_1	-1.506×10^{-3}	7.063×10^{-5}

Table 5.2: Parameter estimates from a binary logistic regression on the observed connectivity data on the sensor devices based on the Texas Instruments hardware.

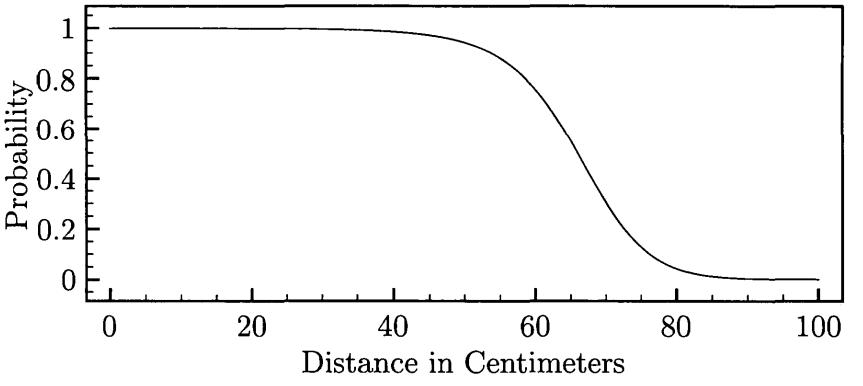


Figure 5.4: The probability of a connection between a pair of nodes in an obstacle free environment according to the empirical disk (ED_{TI}) model.

The communication range r is a parameter of the Unit Disk and Noisy Disk models. The Empirical Disk model is generated from connectivity data and the communication range is not deterministic. As the distance between a pair of nodes increases, the less likely they are to be connected and become neighbours. Consider a node u. The communication range r is needed to impose constraints on the position of node u and should approximately quantify the threshold distance between

the node u and its neighbour nodes $N(u,1)$. The communication range r is an estimate of the communication range of a radio in a sensor device. If the estimate of the communication range r for a node is lower than the actual communication range, then there will be more nodes in $N(u,1)$ where $d(u,v) > r$, $v \in N(u,1)$. If the estimate of the communication range r for a node is higher than the actual communication range, then there will be more nodes in $N(u,1)^c$ where $d(u,w) < r$, $w \in N(u,1)^c$. Both of these conditions violate the connectivity constraints imposed node u. To strike a balance between these two conditions, the communication range r is set to the distance where there is probability 0.5 of receiving a message.

Chapter 6

A Positioning Algorithm for a System of Mobile Nodes

This chapter proposes a new distributed algorithm for positioning called Orbit that has several novel features [2]. A special graph structure is identified that imposes a new set of constraints on sensor positions. It will be shown that this graph property can be found and exploited efficiently by Orbit. A design element of Orbit that is distinct from other mobile positioning algorithms is that it uses cooperation between nearby nodes to improve positional information. Traditionally a node only provides constraint information about itself. Orbit enables a node to identify and communicate the existence of more complex spatial structures which impose new constraints. Some of these new constraints apply solely to other nearby nodes and could not be imposed without a cooperative approach. This is also the first algo-

[2]This work has been published in [36]

rithmic work to characterize different shapes and sizes of the region of uncertainty, the set of feasible positions for a node. It will be shown that, for standard node deployments and mobility models, some of these sets are disjoint which contributes to the propagation of positional error. The Orbit algorithm uses a fast and effective heuristic for handling these disjoint sets. Both the new constraints and the special handling of disjoint sets of positions give significant reductions to positional error.

6.1 Constraints from Node Connectivity

A positional **constraint** is a condition on the Euclidean distance between a pair of nodes. The constraint has the form $\mathfrak{L} \leq d(p_t(u_i), p_t(u_j)) \leq \mathfrak{U}$, where $p_t(u_i)$ and $p_t(u_j)$ are node positions in $\mathbb{R}^2$ at time t and $\mathfrak{L}$ expresses the lower limit on the distance and $\mathfrak{U}$ expresses the upper limit on the distance.

Constraints are imposed on a node by the connectivity graph. Nodes are assumed to know an upper limit r on their communication range so that a limit on the distance between a pair of connected nodes can be defined. For example, if node u_i is connected to node u_j at time t, then the position of node u_i is constrained by the expression $d(p_t(u_i), p_t(u_j)) \leq r$. Similarly, the absence of connectivity between a pair of nodes is also used to impose constraints: if u_i and u_j are not connected then $d(p_t(u_i), p_t(u_j)) > r$. This formulation is based on disk shaped communication regions but the experimentation will demonstrate that Orbit performs well under

more complex communication models.

A node has a set of constraints imposed on its current position. The constraints imposed by a non-seed node u_j must be adjusted by the positional error of its position estimate. Let $e_{x_{q_{t-1}}}(u_j)$, $e_{y_{q_{t-1}}}(u_j)$ be the maximum positional error of the position estimate $q_{t-1}(u_j)$ in the x-dimension and y-dimension, respectively. Since the position estimate was computed in time step $t-1$, the constraints must be adjusted by δ, the maximum movement distance per time step. By definition the positional error of seed nodes is zero. The constraints can easily be adjusted to handled seed nodes with non-zero error.

The following set of base constraints are applicable to a node u_i in time t [67]:

$$0 \leq \quad d(p_t(u_i), p_t(u_j)) \quad \leq r, u_j \in N_t(u_i, 1) \tag{6.1}$$

$$r < \quad d(p_t(u_i), p_t(u_j)) \quad \leq 2r, u_j \in N_t(u_i, 2) \tag{6.2}$$

$$0 \leq \quad d(p_{t-1}(u_i), p_t(u_i)) \quad \leq \delta \tag{6.3}$$

$$0 \leq \quad d(x_{p_{t-1}}(u_i), x_{q_{t-1}}(u_i)) \quad \leq e_{x_{q_{t-1}}}(u_i) \tag{6.4}$$

$$0 \leq \quad d(y_{p_{t-1}}(u_i), y_{q_{t-1}}(u_i)) \quad \leq e_{y_{q_{t-1}}}(u_i) \tag{6.5}$$

Systems of mobile nodes can make use of constraints imposed in previous time steps. If in the previous time step $t-1$ the position of node u_i was constrained by the expression $d(p_{t-1}(u_i), p_{t-1}(u_j)) \leq \mathfrak{U}$, then the position of node u in time step t,

after it has moved, is constrained by the expression $d(p_t(u_i), p_{t-1}(u_j)) \leq \mathfrak{U} + \delta$ in the current time step.

The composition of the constraints on node u_i's position at time t forms a **region of uncertainty**, denoted $\mathcal{R}_t(u_i)$. This set defines all feasible positions for the node [61]. A region of uncertainty is illustrated in Figures 6.1c and 6.2. The shape of such a region can be quite complex and difficult to model. A set of sample points from $\mathbb{R}^2$ can be used to easily describe the region of uncertainty and form the basis for sample-based positioning algorithms.

6.2 Special Structures with Disk Graphs

The region of uncertainty $\mathcal{R}(u)$ is the set of points in $\mathbb{R}^2$ that satisfy all constraints on the position of node u. A novel aspect of the Orbit positioning algorithm is the characterization and application of a special graph structure called a star graph that imposes new constraints on the positions of a set of nodes. The new constraints reduce the size of the region of uncertainty for a node thereby reducing the set of feasible positions for a node.

A **star graph** is a bipartite graph with the form $K_{1,n}$ [14]. The singleton node is the **root node** and the others are **leaf nodes**. A set of nodes is **independent** if no pair of nodes in the set are connected. An independent neighbourhood is a set $I \subseteq N(u_0, 1)$ where u_0 is the root node and the nodes in I are independent.

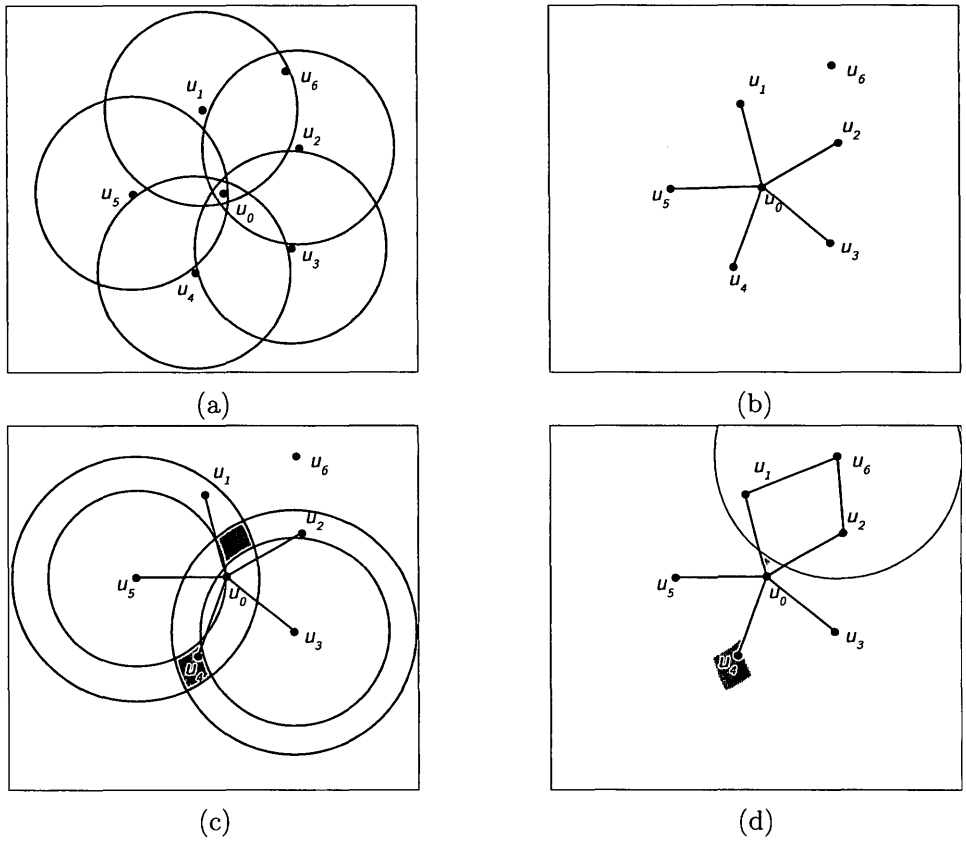


Figure 6.1: An illustration of five independent neighbours I of node u_0 and some of the constraints this structures imposes on the position of node u_4. Figure 6.1a shows node u_0 with a set of independent neighbours $I = \{u_1, \ldots, u_5\}$. The circles represent the communication regions of each neighbour node of u_0. The communication region of each node in I does not contain the position any of the other nodes in I because they are independent. The star graph describing the disk intersection is shown in Figure 6.1b. Figure 6.1c shows the region of uncertainty, shaded grey, for node u_4, when only the constraints imposed by u_3 and u_5 are considered (the cyclic order of the leaf nodes is assumed). These constraints are based on the independence and cyclic order of the two non-adjacent nodes u_3 and u_5. Geometrically, the region is constructed by the intersection of two annuli. The use of negative constraints results in the region being composed of two isolated components. If a third constraint from node u_6 is imposed on the position of node u_4, as illustrated in Figure 6.1d, then the region of uncertainty is composed of two isolated components of unequal size.

Two types of constraints are imposed by the star graph with independent leaf nodes: (i) **root-leaf** constraints and, (ii) **leaf-leaf** constraints. The root-leaf constraints restrict the distance between the root node and a leaf node of the star graph. The leaf-leaf constraints restrict the distance between a pair of leaf nodes in a star graph. Both constraint types are due to independence but the latter requires the leaf nodes to be ordered. The order of the leaf nodes is used to determine the proximity of a pair of leaf nodes.

Lemma 5 (Marathe et al., [39]). *Given a unit disk graph, the maximum number of independent nodes in any neighbourhood is five.*

If $I \subseteq N(u_0, 1)$, then from Lemma 5, the largest set of nodes I in any unit disk graph is five. The subgraph induced by the nodes $I \cup \{u_0\}$ is a star graph, where the nodes in I are leaf nodes and node u_0 is the root node shown in Figure 6.1. The independence property that restricts the maximum size of I also limits the space of realizations of the star graph in $\mathbb{R}^2$. The star graph with five leaves has a tightly constrained form when it is a subset of a disk graph. Throughout the rest of this paper each reference to a star graph implies it is a subgraph of a disk graph.

Definition 3 (Huntington, [23]). *A set of leaf nodes I and a ternary relationship R on these nodes define a* ***cyclic order*** *if the following properties of R are satisfied:*

1. *If $a, b, c \in I$ are distinct, then $(a, b, c) \in R \rightarrow (b, c, a) \in R$*

2. *If* $a, b, c \in I$ *are distinct, then* $(a, b, c) \in R \rightarrow (c, b, a) \notin R$

3. *If* $a, b, c \in I$ *are distinct, then* $(a, b, c), (a, c, d) \in R \rightarrow (a, b, d) \in R$

4. *If* $a, b, c \in I$ *are distinct, then either* $(a, b, c) \in R$ *or* $(c, b, a) \in R$

5. $(a, b, c) \in R \rightarrow a, b, c \in I$ *are distinct*

From the definition of independence, the leaf nodes I are not connected and reside at distinct points. Let be the cyclic order R of the leaf nodes I be given by the clockwise arrangement of the projection of each leaf node position onto the circle of radius r about the position of the root node u_0. The points of a circle satisfy Definition 3 and therefore so do the projection of the leaf node positions.

Definition 4. *Consider a set of leaf nodes* I *that are in cyclic order defined by a relation* R. *A pair of distinct leaf nodes* a, b *are* ***order-adjacent*** *if*

1. $\nexists x \in I$ *such that* $(a, x, b) \in R$, *or*

2. $\nexists x \in I$ *such that* $(b, x, a) \in R$

Lemma 6 is used to put a set of leaf nodes in a cyclic order.

Lemma 6. *Let a star graph be formed by a set leaf nodes* $I = \{u_1, \ldots, u_5\}$ *and a root node* u_0. *If a node* u_6, *where* $u_6 \notin N(u_0)$, *is a common neighbour of exactly two leaf nodes in* I, *then those two leaf nodes are order-adjacent.*

Proof. Consider a pair of leaf nodes $u_1, u_2 \in I$ with a common neighbour u_6 where $u_6 \notin N(u_0)$. Refer to Figure 6.1d for an illustration.

Suppose to the contrary that leaf nodes u_1, u_2 are not order-adjacent. Then there exists another node $u_i \in I - \{u_1, u_2\}$ that is between nodes u_1 and u_2. The leaf node u_i is contained in the triangle $\triangle u_0 u_1 u_2$ since it is between nodes u_1 and u_2 and it is not a neighbour of node u_6. The triangle $\triangle u_i u_1 u_2$ shares the side $\overline{u_1 u_2}$ with the triangle $\triangle u_0 u_1 u_2$ so $d(u_i, u_1) + d(u_i, u_2) \leq d(u_0, u_1) + d(u_0, u_2) \leq 2$. However nodes u_1, u_2, u_i are independent and therefore $d(u_i, u_1) + d(u_i, u_2) > 2$, which is a contradiction. Therefore leaf nodes u_1 and u_2 are order-adjacent. □

Figure 6.1d shows five independent nodes I where leaf nodes $u_1, u_2 \in I$ are order-adjacent.

Once node u knows its first neighbours $N(u, 1)$ and second neighbours $N(u, 2)$, it can determine whether it is the root of a star graph. The restrictions on the embedding of the star graph are translated into tighter constraints between the first and second neighbours. Each leaf node is a first neighbour of the root node and vice versa. If nodes u and v are first neighbours in the star graph, Orbit imposes Constraint 6.6, which is a root-leaf constraint. Each leaf node is a second neighbour of every other leaf node in the star graph. There is a cyclic ordering on the leaf nodes. If two leaf nodes u and v are adjacent in the order then Constraint 6.7 is imposed. Otherwise Constraint 6.8, which is a leaf-leaf constraint, is imposed on

second neighbours.

The star graph yields an enhanced set of constraints between its member nodes. These new constraints are communicated from the root node to the leaves. In this way nodes can cooperate in the positioning process, beyond exchanging their position estimates.

$$\text{Root node } u_0 \text{ and } u_i \in I, \quad 0.7r < d(p_t(u), p_t(v)) \leq r, \tag{6.6}$$

$$\text{Adjacent } u_i, u_j \in I, \quad 1.45r < d(p_t(u), p_t(v)) \leq 2r, \tag{6.7}$$

$$\text{Non-adjacent } u_i, u_j \in I, \quad r < d(p_t(u), p_t(v)) \leq 1.40r \tag{6.8}$$

The values of the lower limit $\mathfrak{L}$ and upper limit $\mathfrak{U}$ in Constraints 6.6, 6.7, and 6.8 were determined from the empirical distribution of distances between nodes in realizations of the star graph. The condition in each expression captures at least 99% of the realizations of a star graph arising from randomly generated embeddings. There are some realizations of a star graph that violate the conditions specified for Constraints 6.6, 6.7, and 6.8 but they are very infrequent. Setting the conditions to cover all realizations of the star graph would add considerable slack to the constraints.

6.3 Orbit Positioning Algorithm

Orbit is a positioning algorithm that reduces positional error by exploiting the existence of star graphs with independent leaf nodes to impose new constraints on node positions and managing the propagation of error due to isolated components composing the region of uncertainty. Both of these features serve to reduce the size of the region of uncertainty for a node. Orbit is a distributed algorithm that proceeds in rounds, one per time step t. Each round has two phases: (i) a communication phase where nodes communicate by broadcasting messages and, (ii) a computation phase where each node performs computations for the purpose of position estimation. The Orbit algorithm adopts the sample-based framework.

6.3.1 Overview of Sample-based Positioning

The general sampling process for a node u_i at time t:

1. Construct sample box $\mathcal{B}_t(u_i)$ using the constraints on the position of node u_i at time t.

2. Generate a set $L_t = \{l_t^1, l_t^2, \ldots, l_t^\eta\}$ of η sample points contained in the rectangle $\mathcal{B}_t(u_i)$. A sample point $l_t^k \in L_t$ is a point in $\mathcal{B}_t(u_i) \subset \mathbb{R}^2$.

3. Remove infeasible (and unlikely) sample points from L_t so that the remaining

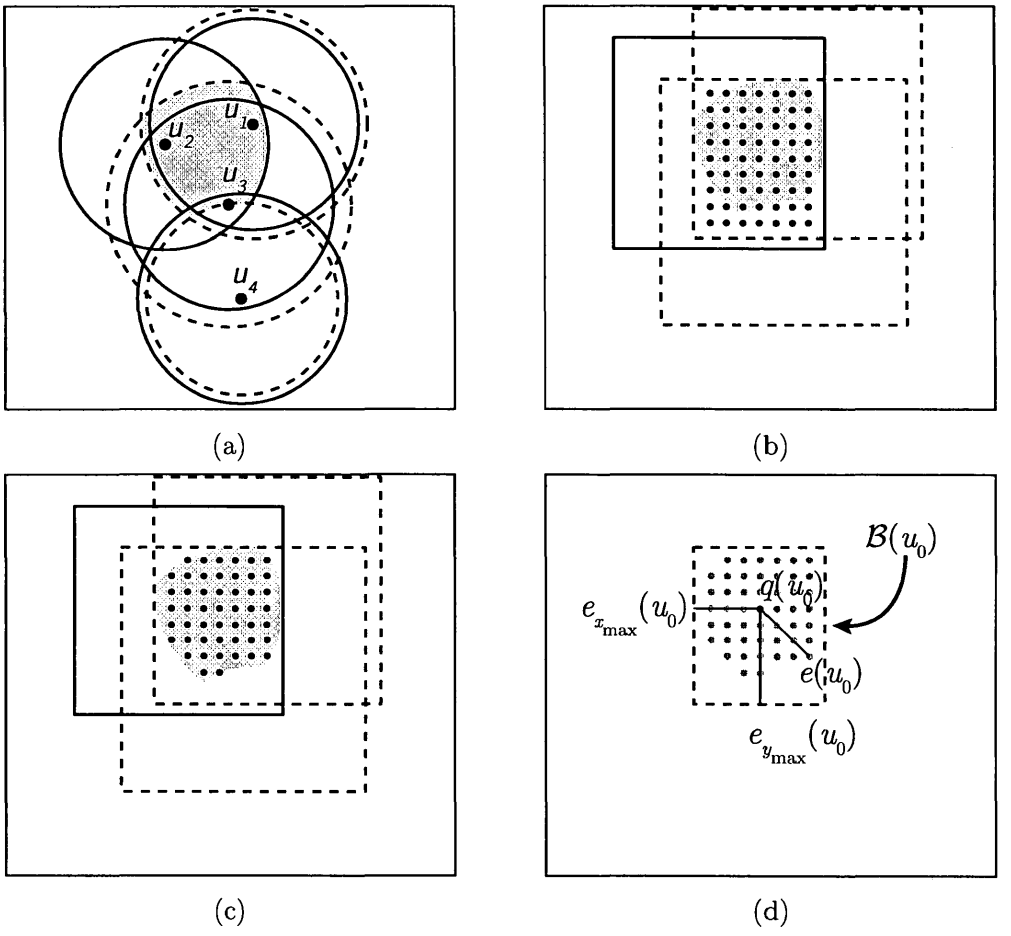


Figure 6.2: An illustration of the process of sampling the region of uncertainty which is composed from constraints. Figure 6.2a shows the positional constraints, geometrically realized as circles, imposed on a node u_0 by the three first neighbour nodes $u_1 \ldots u_3 \in N(u_0, 1)$ and a second neighbour node $u_4 \in N(u_0, 2)$. The solid circle represents the communication region of range r and the dashed circle represents the constraint adjusted for positional error. Node u_2 is a seed node so it has no positional error. Node u_4 has a contracted constraint because it is a negative constraint. The conjunction of the constraints gives the region of uncertainty which is shaded grey. The positions of the sample points in the sample box is shown Figure 6.2b and the remaining sample points after filtering is shown in Figure 6.2c. The positional error estimates $e_{x_{\max}}(u_0)$, $e_{y_{\max}}(u_0)$, and $e(u_0)$ are shown in Figure 6.2d.

sample points are in the region of uncertainty $\mathcal{R}_t(u_i) \subset \mathcal{B}_t(u_i)$.

4. Estimate position $q_t(u_i)$ using the remaining sample points.

All connectivity-based positioning algorithms that use the sample point approach follow these steps.

The sample points are an approximation of the region of uncertainty for a node. The use of a sample box makes the generation of sample points straightforward and the subsequent application of the constraints on the node's position can remove those sample points that fall outside the region of uncertainty. This framework is flexible and has the following benefits: (i) working with sample points does not involve any complex algorithms, (ii) using the sample points it becomes trivial to describe the complex shapes of the region of uncertainty that arise during the positioning process and, (iii) handling the case where nodes are mobile requires only minor adjustments since sample points can easily be associated with time.

During the communication phase each node broadcasts a message containing the identification and positional information about itself and its neighbours. This phase allows each node to learn of the local connectivity information and positional information. The connectivity information is used to impose constraints on each node's position as described in Section 6.1. The message complexity is given in Section 6.3.5.

Algorithm 1 Overview of Orbit on node u_i

1: **if** $I = \emptyset$ **then**
2: $I \leftarrow$ SEARCHFORLEAFNODES($N(u_i, 1)$)
3: **if** $I \neq \emptyset$ **then**
4: ORDERLEAFNODES(I,$N(u_i, 1)$,$\{N(u_j, 1)\}_{u_j \in N(u_i,1)}$)
5: **end if**
6: **end if**
7: $B_t \leftarrow$BUILDSAMPLEBOX($N(u_i, 1), N(u_i, 2)$)
8: **if** $L_{t-1} = \emptyset \vee \delta = 0$ **then**
9: $L_t \leftarrow$ GENERATESAMPLEPOINTS($\mathcal{B}_t$)
10: **else**
11: $L_t \leftarrow$ GENERATESAMPLEPOINTS($\mathcal{B}_t, L_{t-1}$)
12: **end if**
13: $L_t^* \leftarrow$ FILTERSAMPLEPOINTS($L_t, N(u_i, 1), N(u_i, 2), I$)
14: $L_t^{**} \leftarrow$ PARTITIONSAMPLEPOINTS(L_t^*)
15: **if** $L_t^{**} \neq \emptyset$ **then**
16: ESTIMATEPOSITION(L_t^{**})
17: $L_{t-1} \leftarrow L_t^{**}$
18: **end if**

The novel aspects of Orbit are the addition of new positional constraints imposed on node positions and addressing the issue of isolated components that compose the region of uncertainty. Both of these features reduce the size of the region of uncertainty and therefore reduce the positional error of a node's estimated position. The pseudocode for Orbit is presented in Algorithm 1. The novel features of the Orbit algorithm are elaborated on in the following sections. The SEARCHFORLEAFNODES and ORDERLEAFNODES routines impose new constraints based on independent nodes, if they exist, and are described in Section 6.3.3. The GENERATESAMPLEPOINTS routine generates sample points in preparation for approximating the region of uncertainty and is described in Section 6.3.2. The PARTITIONSAMPLEPOINTS

routine, described in Section 6.3.4, handles isolated components that compose the regions of uncertainty. The treatment of the isolated components can reduce the size of the region of uncertainty. The BUILDSAMPLEBOX, FILTERSAMPLEPOINTS, and ESTIMATEPOSITION routines are identical to those proposed in WMCL-B, another recently proposed connectivity-based positioning algorithm [67]. These later routines are discussed below.

The BUILDSAMPLEBOX routine finds a small axis-aligned rectangle $\mathcal{B}_t(u_i)$, referred to as the sample box, that encloses the region of uncertainty. A sample box is illustrated in Figure 6.2. The coordinates of the sample box are computed by iterating over the constraints and recording the extremal points. For example, the smallest boundary point in the x-dimension of the sample box calculated using neighbouring seed nodes is

$$x_{\min} = \max\{x_{p_t}(u_j) - r\}, u_j \in \dot{N}(u_i, 1).$$

The non-seed neighbour nodes and second neighbour seed nodes can be used to reduce the size of the sample box [67]. The reduction in size improves the efficiency of the sampling process because fewer sample points will be discarded during the sample point removal in the FILTERSAMPLEPOINTS routine.

The FILTERSAMPLEPOINTS routine generates the set of sample points L_t^* by removing sample points in L_t that do not reside in the region of uncertainty $\mathcal{R}_t(u_i)$. This is achieved by checking if each sample point satisfies the set of constraints on the node's position. If a sample point does not satisfy a constraint, it is not added to the set L_t^*.

The position of a node u_i is estimated in the ESTIMATEPOSITION routine. This routine calculates the mean of the set of sample points L_t^{**}, the sample points that remain after the handling of potential isolated components in the PARTITIONSAMPLEPOINTS routine, using the following calculations:

$$x_{q_t}(u_i) = \frac{\sum_{l_t^k \in L_t^{**}} x_t^k}{|L_t^{**}|}, y_{q_t}(u_i) = \frac{\sum_{l_t^k \in L_t^{**}} y_t^k}{|L_t^{**}|}, \tag{6.9}$$

where (x_t^k, y_t^k) are the coordinates of sample point l_t^k and $|L_t^{**}| > 0$. The point $q_t(u_i)$ is the position assigned to node u_i and is an approximation of the centroid of the region of uncertainty.

The ESTIMATEPOSITION routine also estimates a node's positional error. The positional error of the estimate is decomposed into x and y components $e_{x_{q_t}}(u_i)$ and $e_{y_{q_t}}(u_i)$. If the coordinates of the bounding box are $(x_{\min}, x_{\max}, y_{\min}, y_{\max})$ then $e_{x_{q_t}}(u_i) = \max\{x_{q_t}(u_i) - x_{\min}, x_{\max} - x_{q_t}(u_i)\}$ [67]. A similar calculation is used for $e_{y_{q_t}}(u_i)$. The positional error calculation is $e_{q_t}(u_i) = \max\{d(l_t^k, q_t(u_i)), l_t^k \in L_t^{**}\}$.

The error values are illustrated in Figure 6.2d.

6.3.2 Generating Sample Points

The GENERATESAMPLEPOINTS routine generates a set of sample points which are arranged on a regular grid as shown in Figure 6.2. The grid is bounded by the sample box $\mathcal{B}$. If a node has no recorded samples points from time step $t-1$ or if $\delta = 0$, then it generates a set of η sample points L_t within the sample box. Otherwise the sample points from time step $t-1$, L_{t-1}, are used to generate the new set of sample points L_t. For each sample point $l_{t-1}^k \in L_{t-1}$, a point p is randomly selected from the disk of radius δ about l_{t-1}^k. If point $p \in \mathcal{B}$ then a new sample point l_t^k is created at nearest grid cell to point p and added to L_t. Otherwise a new sample point is created.

The sample generation process allows a grid cell to contain more than one sample point. This puts greater weight on some locations in the grid which impacts the mean sample point.

6.3.3 Finding Maximal Independent Sets of Nodes

One of the primary features of the Orbit algorithm is the use of new constraints imposed by sets of independent nodes. Section 6.2 describes the new constraints which are applied in the FILTERSAMPLEPOINTS routine. This section will show

how to efficiently find independent sets of size five and place them in cyclic order so that the new constraints may be imposed on node positions. The cyclic order is used to impose leaf-leaf constraints on node positions. Each star graph impacts six nodes with new constraints on their positions.

The frequency of occurrence of nodes that contain an independent sets of five neighbours is shown in Figure 6.3.

Algorithm 2 Searching for the leaf nodes on node u_i

```
1: procedure SEARCHFORLEAFNODES(N(u_i, 1))
2:     for k ← 0 to 4 do
3:         X ← N(u_i, 1)
4:         I ← {v_(k)}
5:         X ← X \ ({v_(k)} ∪ N(v_(k), 1))
6:         while X ≠ ∅ do
7:             I ← {v_(0)} ∪ I
8:             X ← X \ ({v_(0)} ∪ N(v_(0), 1))
9:         end while
10:        if |I| = 5 then
11:            return I
12:        end if
13:    end for
14:    return ∅
15: end procedure
```

The naive algorithm is exhaustive, testing all 5-subsets of the set of neighbours $N(u_i, 1)$ for independence. Although this algorithm runs in polynomial time and the size of the input to this algorithm would be small (usually less than 50 nodes for the networks used in related research), it is considered an expensive operation to execute on a resource limited device.

The routine SEARCHFORLEAFNODES is a greedy approach for searching for an independent set of nodes. The pseudocode for the routine is given in Algorithm 2.

The routine begins by selecting the nodes $u_j \in N(u_i, 1)$ with the smallest number of neighbours in common with $N(u_i, 1)$. This heuristic approach is based on the observation that the neighbours of u_i with the fewest neighbours in common with u_i are usually a greater Euclidean distance from u_i. This heuristic is used to prioritize the search for a set of independent nodes. Neighbours with the smallest number of nodes in common with u_i are sequentially selected and removed from candidacy until there are no more candidates. The search for an independent set of nodes is attempted five times. In each attempt, the search begins by selecting node $v_{(k)}$, the node with the k^{th} smallest number neighbours in common with node u_i. Starting each search attempt with a different node increases the frequency with which an independent set is found.

Property 1. *Consider X, the set of remaining neighbours for leaf node candidacy, and I, the set of independent nodes, computed by Algorithm 2. The nodes in the set X are independent from each node in I.*

The nodes in the set X are the remaining possibilities for leaf node candidacy. Property 1 is guaranteed by the removal of a node and its neighbours from the set X when it is added to I. The set I will contains at most five nodes by Lemma 5.

Therefore, if I contains five nodes the largest set of independent nodes in the neighbourhood has been found. If node u_i finds a set of independent nodes in its neighbourhood the set is broadcast to its neighbours at the end of the communication phase.

Consider the SEARCHFORLEAFNODES routine. Let $V' = N(u_i, 1)$ and (V', E') be the subgraph induced by the neighbourhood of u_i. The set V' becomes known to node u_i by communicating with neighbours. Each execution of the outer loop, beginning on line 2 of the SEARCHFORLEAFNODES routine, requires the degree of each node in V'. The degrees can be calculated by examining the edges incident to each node $v \in V'$ and therefore the calculation requires $O(|E'|)$ operations. Finding the node v with the kth smallest degree requires a linear scan of the degrees of each node. This calculation requires $O(|V'|)$ steps. Removing the node v and each of its neighbours from X requires at most $O(|V'|)$ steps. After each removal, the degree of each node is adjusted. The inner loop, beginning on line 6 of the SEARCHFORLEAFNODES routine, is executed at most five times by Lemma 5 and Property 1. Therefore the running time of SEARCHFORLEAFNODES is $O(|V'| + |E'|)$.

The quality of the SEARCHFORLEAFNODES routine was tested with a computer simulation. If an independent set of five nodes exists, the routine finds the set more than 97% of the time.

The ORDERLEAFNODES routine searches for a cyclic ordering of the leaf nodes. Let u_i be a root node of a star graph with leaf nodes I. The routine constructs a graph (I, E') where I are the five leaf nodes and the edges E' represent order-adjacent leaf nodes. An edge $(u_j, u_k) \in E'$ if $(N(u_j, 1) \cap N(u_k, 1)) \setminus N(u_i, 1) \neq \emptyset$, where $u_j, u_k \in I$. If $|E'| = 5$, then by Definition 4 and Lemma 6 there is a cyclic ordering of the leaf nodes I.

Let node u_i be the root node and I be the set of five leaf nodes. The SEARCHFORLEAFNODES and ORDERLEAFNODES routines use set union, set intersection and set difference for some of their operations. The size of the independent set of nodes I is five so there are a fixed number of set operations performed to find an ordering of the leaf nodes. The set operations can be computed in a time linear in the size of the input sets once the nodes are ordered by the node identifiers. The running time of the ORDERLEAFNODES routine is linear in the size of the largest neighbourhood $N(u_j)$, $u_j \in I$ when the nodes are sorted by node identifier.

6.3.4 Disconnected Regions of Uncertainty

In Section 6.1 it was shown that the use of negative constraints can cause the region of uncertainty to become composed of isolated components. Figure 6.4 illustrates the positional information of a node captured during a run of the Orbit algorithm in a computer simulation. The negative constraints shown in the figure are effec-

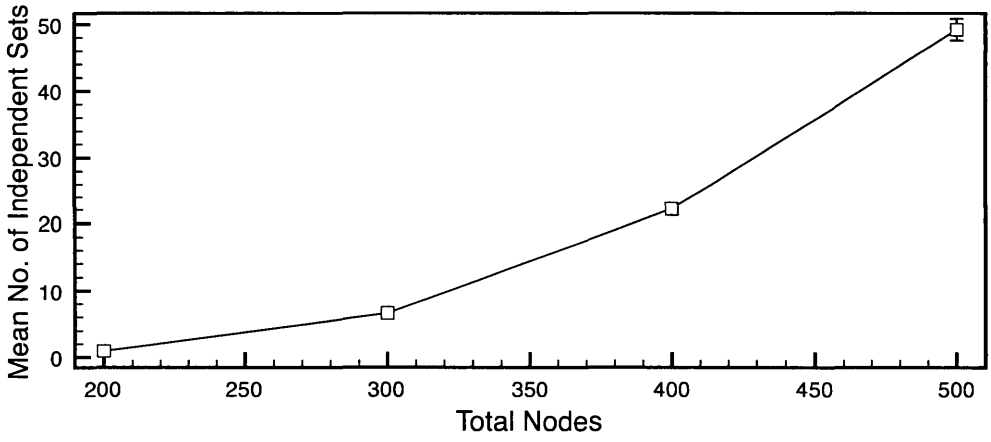


Figure 6.3: Mean number of star graphs in a system of nodes that are uniformly distributed and operating under the unit disk communication model. The relationship between density of nodes and the mean number of star graphs is super-linear. The error bars in represent a 95 percent confidence interval but are often not visible because the interval is so small.

tive at reducing the size of the region of uncertainty but they cause the region of uncertainty to become composed of isolated components.

A node u sends the information $q_{t-1}(u_i)$ and $e_{q_{t-1}}(u_i)$ to its neighbours $N_t(u_i)$ at time t. The fact that $e_{q_{t-1}}(u_i)$ is large does not imply the size region of uncertainty is large. For an example, see the isolated components in Figure 6.4. The region of uncertainty illustrated in the figure is composed of isolated components. The neighbours of node u may use $q_{t-1}(u_i)$ and $e_{q_{t-1}}(u_i)$ to form constraints on their their own positions. A constraint that is based on $q_{t-1}(u_i)$ and $e_{q_{t-1}}(u_i)$ is given in Expressions 6.4 and 6.5. Geometrically, this constraint can be realized as the disk $\mathcal{D}(q_{t-1}(u_i), e_{q_{t-1}}(u_i))$. However this disk can be a poor approximation of the region of uncertainty $\mathcal{R}_{t-1}(u_i)$ of node u_i. The position $p_{t-1}(u_i)$ of node u_i is not contained in the set of positions $\mathcal{D}(q_{t-1}(u_i), e_{q_{t-1}}(u_i)) \setminus \mathcal{R}_{t-1}(u_i)$ and this set can be large when the region of uncertainty $\mathcal{R}_{t-1}(u_i)$ is composed of isolated components. While node u_i can approximate the non-convex region $\mathcal{R}_{t-1}(u_i)$ with its set of sample points L^*_{t-1}, the neighbours $N_t(u_i)$ of node u_i have only the disk $\mathcal{D}(q_{t-1}(u_i), e_{q_{t-1}}(u_i))$. This is problematic because the benefit of the negative constraints was to reduce the size of the region of uncertainty but the disk $\mathcal{D}(q_{t-1}(u_i), e_{q_{t-1}}(u_i))$ does not accurately reflect this region. If a neighbour node $u_j \in N_t(u_i)$ of u_i uses $q_{t-1}(u_i)$ and $e_{q_{t-1}}(u_i)$ to constrain its own position, it can increase the error of the position estimate $q_t(u_j)$ because it assumes the position $p_{t-1}(u_i)$ of node u_i is contained in

the disk $\mathcal{D}(q_{t-1}(u_i), e_{q_{t-1}}(u_i))$.

In addition to contributing to the propagation of positional error, a region of uncertainty composed of isolated components can have undue influence on the position estimate $q_t(u_i)$ of node u_i if the distance between the connected components is large. It has been observed that when a component of the region is less than one-third of the largest component, the true position of the node is unlikely to be located in this component and that the smallest component can be a large distance from largest component.

The PARTITIONSAMPLEPOINTS routine operates on the set of samples L_t^* after the FILTERSAMPLEPOINTS routine has been executed. At this stage all sample points reside in the region of uncertainty. The Orbit algorithm characterizes the isolated components of the region of uncertainty by partitioning the sample points into distinct sets of sample points that are associated with each component. The sample points associated with isolated components that have a very low probability of containing the position of a node are removed from L_t^*.

A simple algorithm is used to determine the isolated components to which sample points belong. The set of sample points reside on a regular grid. Two sample points are connected if they are contained in neighbouring grid cells. The routine performs a depth first search starting at each sample $l_t^k \in L_t^*$ that does not already belong to a component. Any sample points that are connected to l_t^k are marked

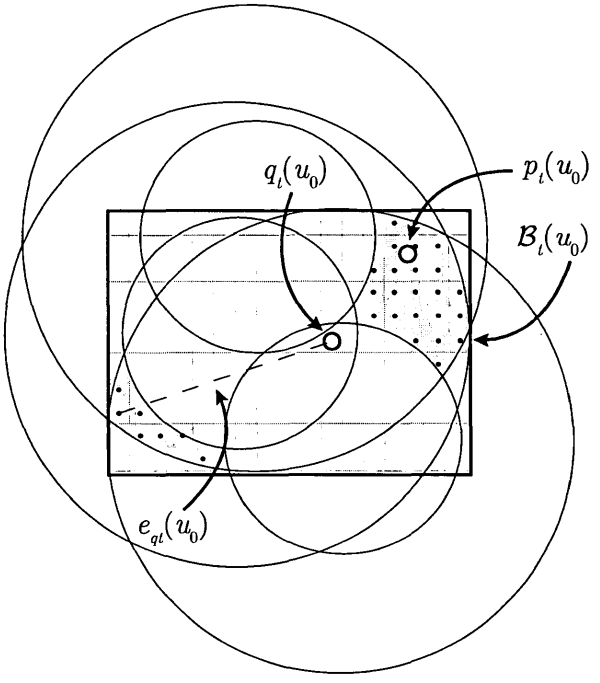
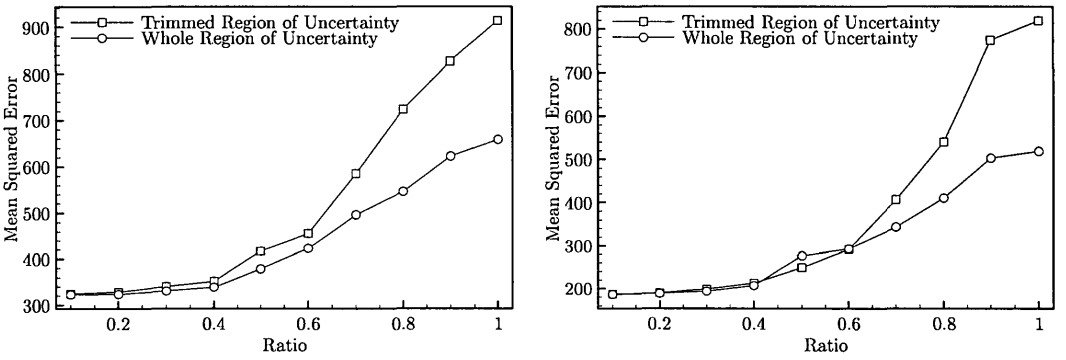


Figure 6.4: The positional information available to node u_0 at time t. The information was captured from a state in the computer simulation of the Orbit positioning algorithm. The large and small disks are the geometric realization of the constraints imposed on the position $p_t(u_0)$ of node u_0. There are six constraints: three upper limits, represented by the large circles, and three lower limits, represented by the small circles. These constraints were used to build the sample box $\mathcal{B}_t(u_0)$. The collection of black dots represents the set of sample points L_t^* that remain after the filtering step. The sample points are aligned to a regular grid which is part of the sample point generation process. Each sample point l_t^k resides inside all large disks and outside all small disks. Therefore each sample point is contained inside the region of uncertainty: $l_t^k \in \mathcal{R}_t(u_0) \subset \mathcal{B}_t(u_0)$, since the sample points that do not satisfy all six constraints are removed. The region of uncertainty is composed of two isolated components which partition the sample points into two connected sets. The size of a connected set is a measure of the size of the region it describes. The position $p_t(u_0)$ of node u_0 is contained in the larger of the two regions. Note the number of sample points used during this particular run of Orbit was $\eta = 500$ but fewer points are shown to simplify the illustration.

and included in the set. The search process stops when no further sample points exist. If there exists an unmarked sample point l_i^j, the process is initiated again with a new set. Each sample point is examined once so this routine runs in a time linear in the total number of sample points η.

The number of sample points in a region is a measurement of the size of that region. The size of the largest set is compared to the other sets. If any set is less than one-third the size of the largest, then the set and its member sample points are discarded. This process trims the region of uncertainty which results in a single connected component.

A computer simulation was used to evaluate the error of position estimates when the region of uncertainty is disconnected. For each disconnected region of uncertainty, two position estimators were evaluated: the centroid of the whole region of uncertainty and the centroid of the trimmed region of uncertainty. The centroid is the position estimate for the node. Figure 6.5 plots the mean squared error of the two estimators which reveals that trimming the smaller component when it is less than one-third the size of the larger component has a minimal effect on the error. However, the size of the trimmed region can result in a reduction in the maximum positional error.



(a) 200 Nodes

(b) 500 Nodes

Figure 6.5: The mean squared error of two position estimators: the estimator that calculates the centroid of the whole region of uncertainty and the estimator that calculates the centroid of the trimmed region of uncertainty.

6.3.5 Message Complexity

The Orbit algorithm uses node identifications to identify the leaf nodes. A node identifier can be encoded in $\lceil \log_2(n) \rceil$ bits, where $n = |V|$, the number of nodes in the network. The positional information for a node is its estimated position $q_t(u_i)$ and the error $e_t(u_i)$. The number of bits to store this information will vary depending on the number of bits representing a number. The positional information is communicated to the neighbourhood of node u_i via a message broadcast. Let k be the total number of bits required to represent the positional information for a node. The Orbit algorithm sends positional information about all neighbours which requires a total of $|N(u_i, 1)| \times (\lceil \log_2(n) \rceil + k)$ bits. Finally, if a node has an independent set of neighbours I it sends the identifications of the nodes in I

which requires $5 \times \lceil \log_2(n) \rceil$ bits. The total number of bits per node per round is $(|N(u_i, 1)| + 1) \times (\lceil \log_2(n) \rceil + k) + 5 \times \lceil \log_2(n) \rceil$.

6.4 Simulation Experiments

A set of computer simulation experiments are conducted to evaluate the performance of the Orbit positioning algorithm. The performance of Orbit is compared to another positioning algorithm, WMCL-B, under a variety of communication and mobility models for a range of parameter values. WMCL-B was selected for the comparison because it is a recently proposed positioning algorithm and it achieves the lowest positional error compared to existing positioning algorithms for mobile systems [67].

There are four primary differences between Orbit and WMCL-B: (i) Orbit uses no knowledge of the field in which the nodes are embedded, unlike WMCL-B which restricts the region of uncertainty to be contained with the boundaries of the field, (ii) Orbit uses a regular grid rather than a purely random process for generating sample points, (iii) Orbit uses new constraints based on independent sets of nodes, (iv) Orbit makes a distinction between a region of uncertainty composed of a single component versus a region of uncertainty composed of isolated components, and processes the two types of region differently.

6.4.1 Methodology

The parameter values for the algorithms in the simulation are identical and both algorithms used a maximum of $\eta = 50$ sample points to approximate the region of uncertainty. For each mobility model, communication model and parameter set, 30 simulations are performed.

The field for the system of nodes V is a bounded Euclidean plane with a dimension of 500×500 units. The star graph structure emerges when approximately 200 or greater nodes are present in the field. Figure 6.3 shows that the number of star graphs increases from 1% to 10% for networks ranging from 200 to 500 nodes.

A mobility model is used to generate a sequence of 200 embeddings $p_0(\cdot) \ldots p_{199}(\cdot)$, one for each time step. Each embedding $p_t(\cdot)$ corresponds to a point in time t that gives the position $p_t(u)$ of each node $u \in V$ in the system. The random walk and random waypoint mobility models have different dynamics. The initial point distribution for the random walk model (RW) is uniform and since the random walk model is a stationary stochastic process the point distribution remains a uniform distribution as the process evolves [55]. The random waypoint model (RP) is not a stationary process but does eventually converge to a stationary distribution [45]. The stationary state of this process is not a uniform point distribution. To remove the effects of the evolving distribution and to speed up the simulation, the initial

point distribution for the random waypoint model is sampled from the stationary distribution [45].

The frequency distribution of the node positions under the Random Walk and Random Waypoint models are shown in Figure 6.6 and Figure 6.7, respectively. The distribution describes the frequency with which a node occupies a small region of the space.

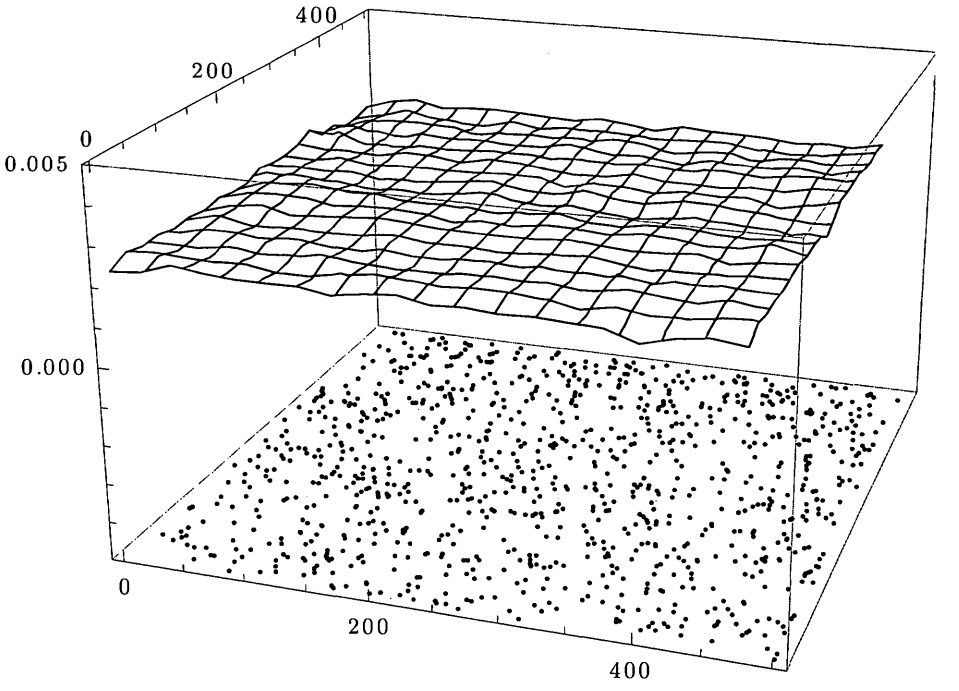


Figure 6.6: The node distribution on the Random Walk model. Below the distribution is a sample of the node positions after 200 steps of movement.

The data for each distribution was created by a computer simulation. The simu-

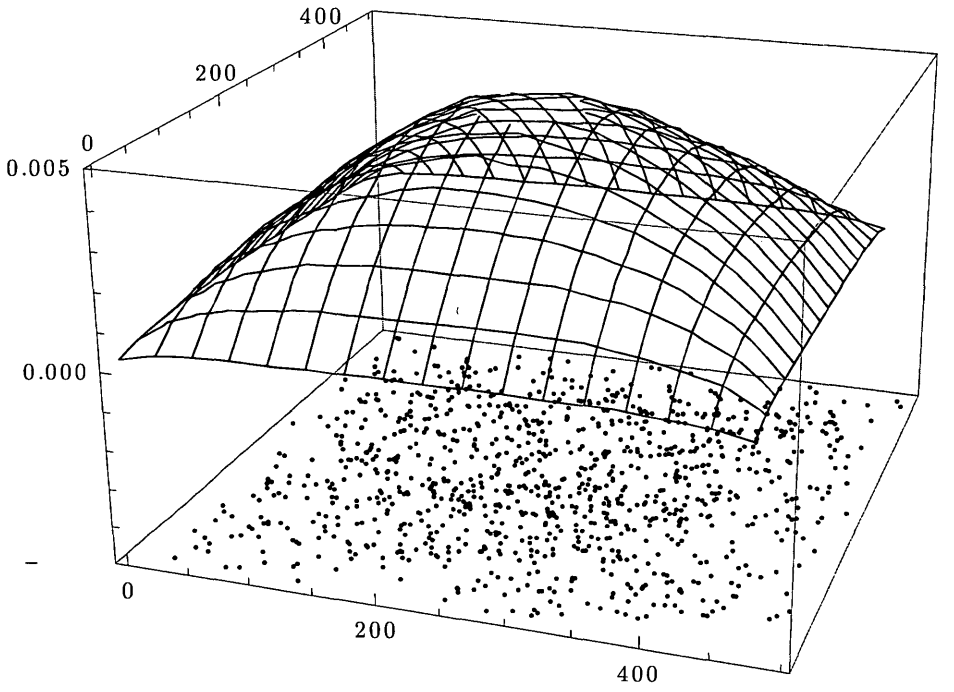


Figure 6.7: The node distribution on the Random Waypoint model. Below the distribution is a sample of the node positions after 200 steps of movement.

lation captured the node positions of 1000 nodes embedded in a 500×500 field after 200 movements. The initial node distribution for each simulation was uniformly random and the node speed was $\delta = 5$. The dynamics of the Random Walk model maintains the homogeneous distribution of nodes as discussed in Chapter 4. When the node movement is governed by the Random Waypoint model, the distribution becomes skewed towards the centre of the space in which the node are embedded. This property of the Random Waypoint model has the effect of increasing the node

density in the centre of the space and decreasing the node density in regions closer to the boundaries of the space. In each graph an embedding of the 1000 nodes is plotted below the distribution. The embedding in each figure was captured after 200 steps of movement.

The information used by Orbit consists of positional constraints imposed by node connectivity. Two factors that affect the connectivity of a pair of nodes are the inter-node distance and radio characteristics. This works considers three communication models to study the impact of connectivity on the performance of each positioning algorithm. The unit disk model (UD) is an idealized radio model, widely used to analyze wireless communication [60]. The simulation uses a radio range $r = 100$ units which connects two nodes u and v at time t if $d(p_t(u), p_t(v)) \leq r$. The noisy disk model (ND) generalizes the unit disk model by adjusting the radio range in each direction according to a single parameter α. To complement these models the probabilistic model empirical disk ($\mathrm{ED_{TI}}$) for communication is also developed from the communication properties of wireless sensor devices. The sensor devices are the Texas Instruments EZ430-RF2500 [24]. The development of this probabilistic model is given in Section 7.1. These latter two models generate communication regions which are irregular and allow for connected nodes that are separated by a distance greater than r, or non-connected nodes that are separated by a distance less than r.

6.4.2 Performance Evaluation

The mean positional error is used as the primary performance measurement. In this section the mean positional error is reported for Orbit and WMCL-B for a variety of model parameter values. The performance of Orbit and WMCL-B is depicted in Figures 6.8, 6.9, 6.10, and 6.11. The error bars in each figure represent a 95 percent confidence interval but are often not visible because the interval is so small. The parameters that are typically reported as affecting performance are (i) the number of seed nodes, (ii) the node density, and (iii) the node speed. The performance of WMCL-B closely corresponds to reported results [67].

Number of Seed Nodes

The mean positional error of both Orbit and WMCL-B improve with more seed nodes in the system, as shown in Figure 6.8. When there are five seed nodes the mean positional error of Orbit is significantly better under the random waypoint model than the random walk model. The dynamics of the random walk model is such that node positions over time are close to their originating position at time $t = 0$. This property is not true of the random waypoint model. Inspection of the embeddings from simulations where the algorithms had the worst positional error revealed that some seed nodes were nearly coincident. Such a configuration reduces

the effectiveness of the seed nodes and this state of near coincidence can persist under the random walk model.

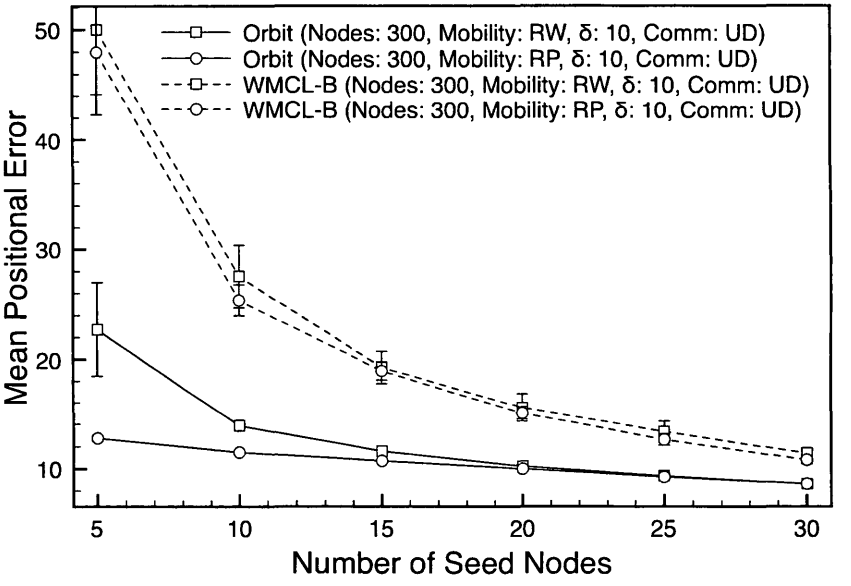


Figure 6.8: Mean Positional Error v. Number of Seeds.

Node density

The impact of node density with the various communication models is shown in Figure 6.9. Under the unit disk model the positional error is decreasing as the number of nodes increases. The positional error is unchanged despite the irregularities in node connectivity due to the noisy disk and empirical disk models. The performance of Orbit exceeds WMCL-B in each model.

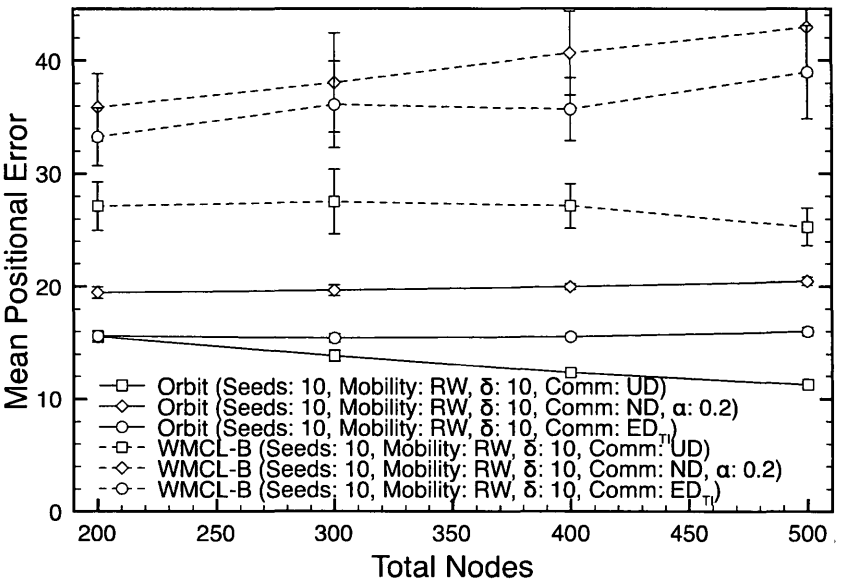


Figure 6.9: Mean Position Error v. Total Nodes.

Node Speed

The impact of node speed is shown in Figure 6.10. At low speeds the performance of the Orbit algorithm is considerably better than WMCL-B. As the node speed increases, the mean positional error of Orbit approaches that of WMCL-B. The increase in positional error is a result of less useful constraint information from past time steps caused by the adjustment of constraints by a larger maximum movement distance δ. The node speed can increase to a point where the past constraints are no longer effective at reducing the region of uncertainty.

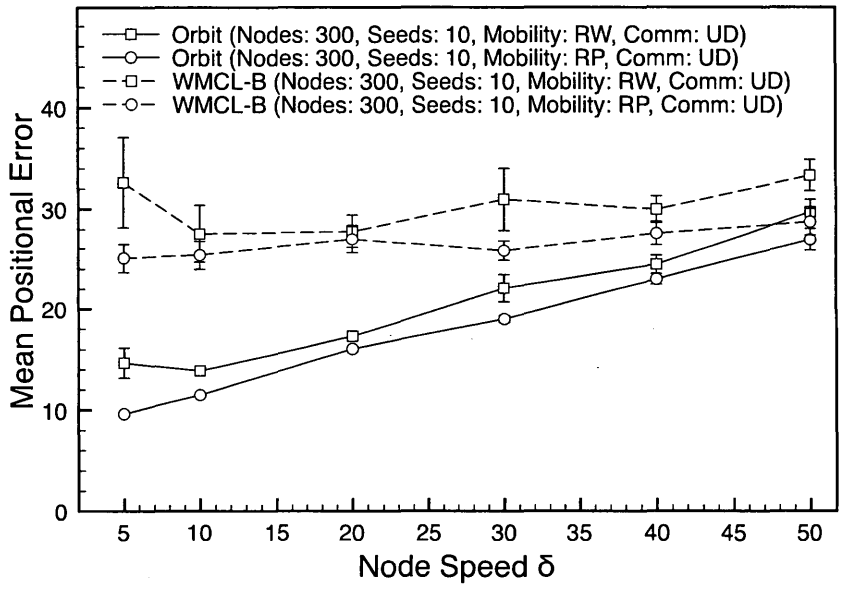


Figure 6.10: Mean Position Error v. Node Speed.

Convergence Time

The mean positional error at each of the first 50 times steps is shown in Figure 6.11. The figure reveals that the Orbit algorithm converges quickly and that the positional error of each algorithm is smaller for connectivity graphs generated by the random waypoint model. A benefit of the faster convergence is discussed in Section 7.4.

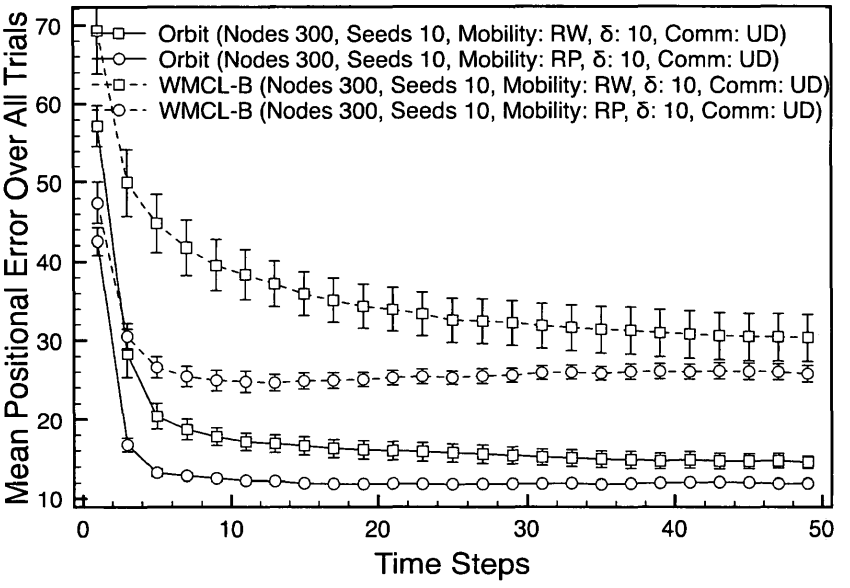


Figure 6.11: Mean Position Error v. Time Step.

Chapter 7

Implementing the Orbit Algorithm on Sensor Devices

To complement the computer simulation an embedded version of Orbit has been developed and executed on resource constrained sensor devices. The devices are composed of low cost components. The purpose of a sensor implementation is twofold:

- Determine if Orbit can be realized on sensor devices
- Validate the positioning results of Orbit achieved on the sensor network with those achieved in a computer simulation

The implementation of Orbit demonstrates the feasibility of the algorithm for real application. Many proposed positioning algorithms are computationally expen-

sive and it is not obvious they can be implemented on resource limited devices. The Orbit algorithm has been executed on simulated networks, as described in Chapter 6, but it is also executed on networks of stationary sensor devices deployed in an environment. The accuracy of the modelling and simulation framework is evaluated by comparing the results of the algorithm on the sensor network with the results of the algorithm in an identical simulated network.

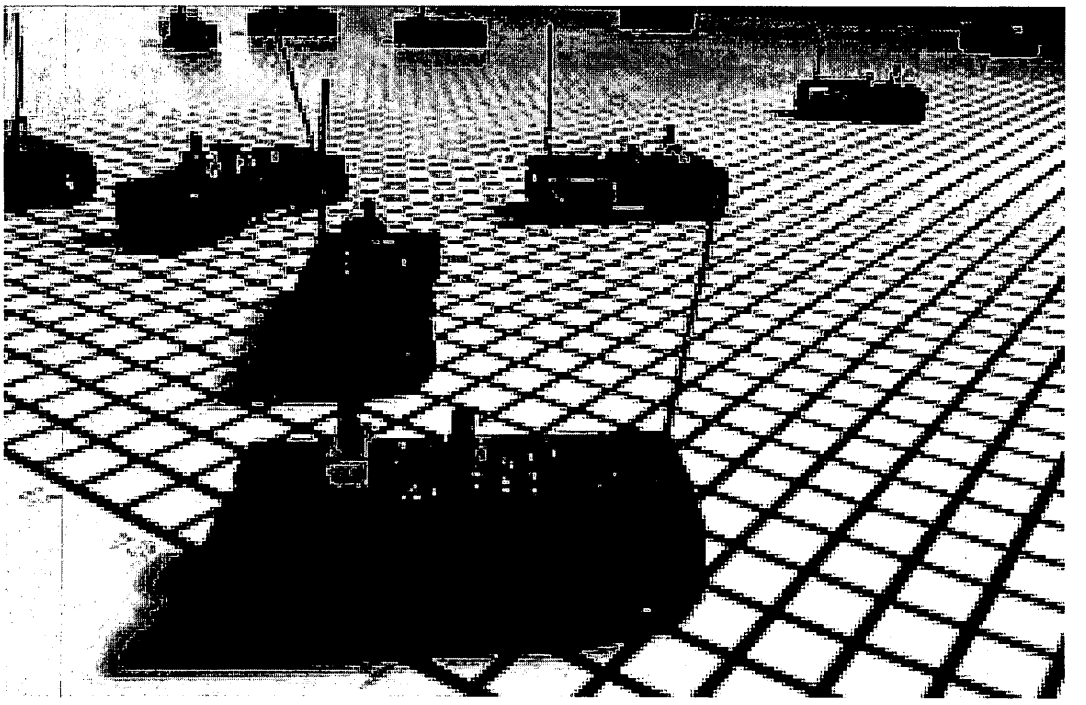


Figure 7.1: Sensors placed within the a bounded two-dimensional sensor field. The units of measure are centimetres.

7.1 Methodology

The sensor devices used for the implementation of Orbit each contain the Nordic nRF24LE1 microcontroller [54]. The critical specifications of the microcontroller are: (i) an 8-bit CPU running at a rate of 16 megahertz with 1 kilobyte of random access memory; (ii) 16 kilobytes of program memory; and (iii) a 2.4 gigahertz radio with a data rate of 2 megabits per second. A monopole antenna is attached to the device and the transmission power of the radio was set to -12dBm to reduce its typical communication range to under a meter. The radio hardware does not provide a measurement of a received signal's strength. A photograph of the device is shown in Figure 7.1.

The testbed for the wireless sensor networks is a two-dimensional regular grid located in an obstacle-free indoor environment which can be seen in Figure 7.1. The position experiments performed with the devices are executed on networks of 15 nodes due to the limited number of sensor devices.

Four embeddings $p_a(\cdot), p_b(\cdot), p_c(\cdot)$, and $p_d(\cdot)$ were generated for evaluating the performance of Orbit. Each embedding of sensor nodes is randomly generated and the nodes with identifications 1 through 3 are seed nodes that know their position. The embeddings were conditionally generated to ensure that two of the embedding, $p_a(\cdot), p_b(\cdot)$, contain an independent set of five nodes and two of the embeddings,

$p_c(\cdot), p_d(\cdot)$, do not contain an independent set of five nodes. Some authors place the seed nodes on the boundary of the space in which the nodes reside which gives a considerable reduction in the positional error of position estimates [67]. The seed node positions in this work are completely random which yields a more general network deployment. Under the unit disk communication model, each embedding induces a network of with a diameter between 2 and 4.

The premise for connectivity-based positioning is that there is a limit r on the communication range of each device and that this limit is shared by all devices. This limit is used to impose constraints on node positions. In reality these assumptions are difficult to establish. The radio experiments described in Section 5.1 was used to determine an appropriate value for the communication range r.

The Orbit positioning algorithm is run once for each embedding. The algorithm uses $\eta = 25$ sample points to approximate the region of uncertainty. The algorithm is run for $T = 10$ rounds after which it is terminated and each node periodically broadcasts its position estimate and neighbourhood. A wireless sink node is used to capture the position estimate from each node.

All tests with the sensors were conducted in an indoor open space at a temperature in the range of 20 °C to 25 °C and a relative humidity in the range of 50% to 65%.

A basic time division multiple access (TDMA) algorithm is used to schedule

communication between nodes. This mechanism is also used to keep each node synchronized to the same round.

Each embedding is also used to generate a connectivity graph under the unit disk (UD) and the empirical disk (ED_N) communication models. Each connectivity graph was input into the simulation experiment of Orbit. The results of the computer simulation and the sensor positioning experiments are compared to evaluate the performance of Orbit and the accuracy of our communication models.

7.2 Software Implementation

The original implementation of the Orbit positioning algorithm was for the computer simulation and was written in the Java programming language. The two most common programming languages available for the sensor devices based on the nRF24LE1 MCU are assembly language and the C language. The latter option was selected for several reasons: (i) speed of development, (ii) ease of debugging. The primary limitations of a sensor device are the available memory and power.

The first challenge in creating an implementation of Orbit for the sensor devices is the limited available memory. Though positioning is the only application of this sensor network, one typically must leave memory for additional software in support of other applications. The implementation of Orbit is designed for the fixed memory configuration of the devices. To simplify the programming effort, the

implementation uses static memory allocation. The final size of the implementation of the Orbit software system is approximately 10 kB of program memory.

An important design criteria for a sensor network is longevity of the sensor network, which means the rate of energy usage must be as small as possible. This is achieved by increasing the duration of the sleep cycles of the sensor devices. In the sleeping state a device draws nearly zero electric current. The sensor nodes have their clocks synchronized so they can have a common sleep schedule during periods of inactivity.

7.3 Sensor Positioning Experiment

The positional error of each node for each embedding $p_a(\cdot), p_b(\cdot), p_c(\cdot)$, and $p_d(\cdot)$ is illustrated in Figures 7.3a, 7.3b, 7.4a, and 7.4b, respectively. A star graph was found by Orbit in both embeddings $p_a(\cdot)$ and $p_b(\cdot)$.

Each embedding was used to generate a connectivity graph under the unit disk (UD) and empirical disk (ED_N) communication models. For comparison with the performance of Orbit on the sensor devices, the simulation experiment was used to execute Orbit on the connectivity graphs for each embedding. The mean positional error and standard deviation over all nodes is presented in Table 7.1. The statistics show that the results for Orbit on the sensor network are similar to results for Orbit in computer simulations using both the unit disk and empirical disk communication

models.

The neighbourhood information from the positioning test was used to determine the relative frequency of node connectivity. The relative frequency of connectivity at various inter-node distances is shown in Figure 7.2 along with the probability distribution for the ED_N communication model. At distances less than 40 cm there is less connectivity than the logistic model predicts and at distances greater than 40 cm there is more connectivity than the logistic predicts. These deviations might be attributed to the sensor devices acting as barriers to communication, communication range differences in the sensor devices, or small blind spots in the communication region that were undetected in the radio experiment described in Chapter 5.

The examination of the node embeddings illustrated in Figures 7.3a, 7.3b, 7.4a, and 7.4b and the statistics on the position estimates reported in Table 7.1 leads one to the following observations: (i) well distributed seed nodes are important for good positional accuracy; (ii) in the absence of good seed placement, the existence of the independent set structure is beneficial to positional accuracy and; (iii) overall, the ED_N communication model is a choice for a simulation experiment that will better predict the mean positional error of Orbit on the sensor devices.

Although the seed nodes were more evenly distributed in the the $p_a(\cdot)$ embedding, the performance of Orbit on the $p_b(\cdot)$ embedding was better because fewer of

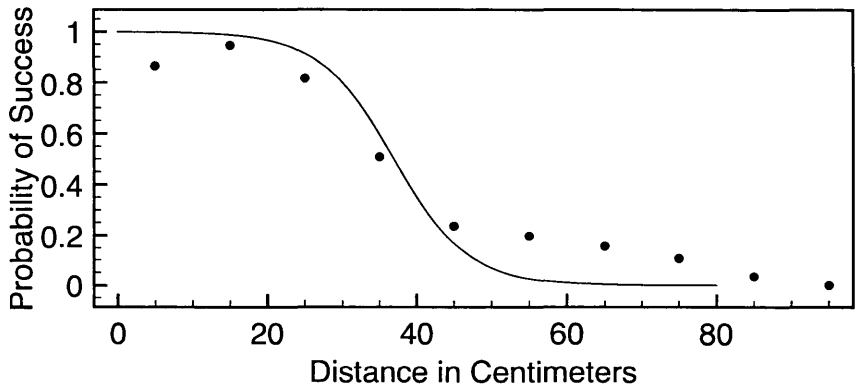


Figure 7.2: The probability of a connection between a pair of nodes in an obstacle free environment according to the empirical disk (ED_N) model. The dashed vertical line at $x = 40$ indicates the selected communication range. The solid dots represent the relative frequency of connectivity derived from the neighbourhood information from the sensor positioning test.

the seed nodes belonged to the star graph. The constraints from the star graph become less useful as the number of seed nodes in the star graph increase because the distance between pairs of seed nodes is known exactly. The performance of Orbit on the nodes in the $p_c(\cdot)$ embedding was relatively poor due to the collinearity of the seed nodes. The position estimates are drawn towards the line of seed nodes. The positional accuracy was good on the nodes in the $p_d(\cdot)$ embedding due to the well distributed seed nodes.

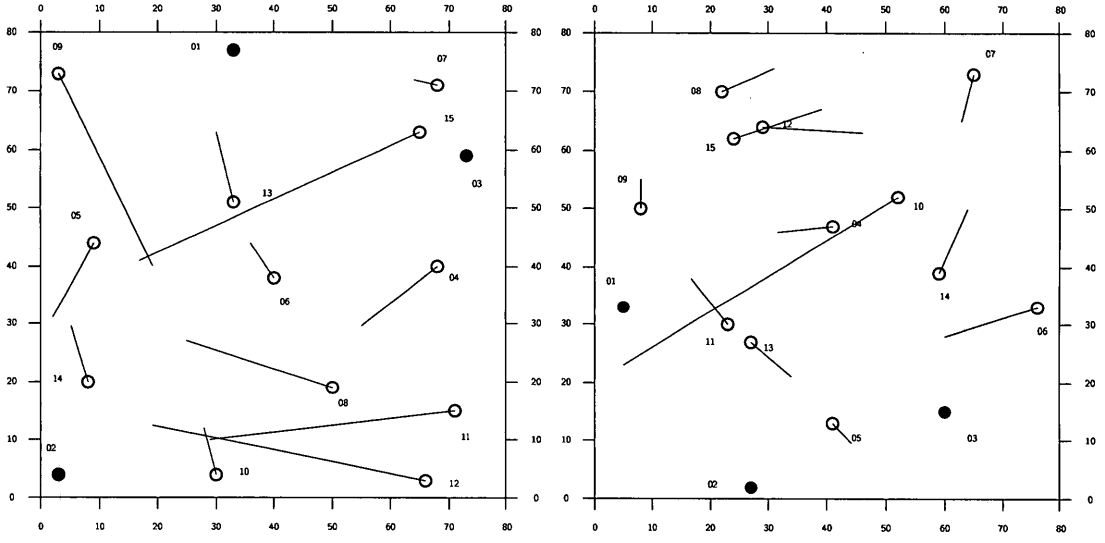
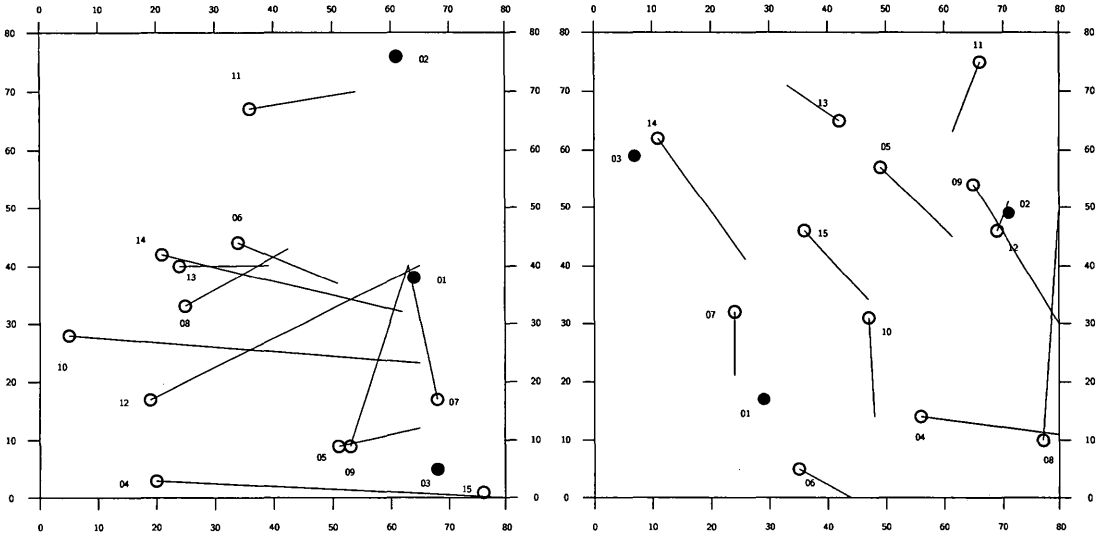


Figure 7.3: Embeddings p_a and p_b that contain an independent set of five nodes under the Unit Disk communication model. The seed nodes are represented by the solid dots and the non-seed nodes are represented by open dots. For each node u_i a line is drawn from the true position to the estimated position of the node. The length of the line represents the node's positional error. For embedding p_a the root node was 06 and the independent set of nodes were $\{01, 03, 05, 10, 11\}$. For embedding p_b the root node was 04 and the independent set of nodes were $\{01, 05, 06, 07, 08\}$.



(a) (b)

Figure 7.4: Embeddings p_c and p_d that do not contain an independent set of five nodes under the Unit Disk communication model. The seed nodes are represented by the solid dots.

Positioning Test	Mean	St. Dev.
Embedding p_a, Simulation (UD)	20.52	10.56
Embedding p_a, Simulation (ED_N)	20.04	11.08
Embedding p_a, Sensor Network	21.28	14.78
Embedding p_b, Simulation (UD)	13.42	9.48
Embedding p_b, Simulation (ED_N)	15.18	7.88
Embedding p_b, Sensor Network	13.99	12.25
Embedding p_c, Simulation (UD)	35.91	16.71
Embedding p_c, Simulation (ED_N)	35.40	18.79
Embedding p_c, Sensor Network	31.17	16.91
Embedding p_d, Simulation (UD)	12.10	10.04
Embedding p_d, Simulation (ED_N)	21.65	14.50
Embedding p_d, Sensor Network	19.24	10.58

Table 7.1: Comparison of the positional error between the computer simulation and the sensor positioning experiment. The computer simulation used the Unit Disk (UD) model for node communication. The mean positional error and standard deviation is calculated over all non-seed nodes.

7.4 Cost of the Most Expensive Routines

The Orbit algorithm has been implemented on resource constrained devices. The practicality of executing an algorithm on such a device is one important factor in determining an algorithm's usefulness in applied settings. One of the possible performance measures of an implementation on a device would be the amount of time required to execute the algorithm, since the device cannot be in a low power state during this time. When a device is in a low power state the amount of current consumed is nearly zero. The selection of an algorithm involves a tradeoff between the quality of the results and the energy required to compute the result. While

more computation might be expected to reduce positional error this could be at a great expense to battery life. Table 7.2 gives the amount of time in microseconds the most expensive routines in Orbit take to completely execute on a sensor device with 30 neighbours. The total time required for computation in Orbit is less than 10 milliseconds on a sensor device.

In addition to computation, distributed positioning algorithms exchange messages with other nodes. The amount of data exchanged between nodes also has an impact on battery life. The message complexity of Orbit and WMCL-B are slightly different. A node running the Orbit algorithm sends information about all of its neighbours in each round while a node running WMCL-B only sends information about the seed nodes in its neighbourhood.

A sensor running the Orbit algorithm requires less than two milliseconds at 2 megabits per second to broadcast the data encoding its positional information if it has 30 neighbours.

The positional information sent by nodes running Orbit is larger than WMCL-B and therefore will yield a higher average current draw. However, examining Figure 6.11 it can be seen that the Orbit algorithm converges to a given level of positional error more quickly than the WMCL-B algorithm. When the nodes reach a desired level of positional error they can enter a sleep state where the typical current consumption is nearly zero. The faster convergence of Orbit means

a network of nodes running this algorithm can complete the positioning task more quickly and move to a sleep state which significantly reduces its average current consumption.

Subroutine	Execution Time (μs)
SEARCHFORLEAFNODES	2004.75
ORDERLEAFNODES	909
GENERATESAMPLEPOINTS	993.75
PARTITIONSAMPLEPOINTS	4488.75

Table 7.2: The mean worst-case time to execute each routine used in the Orbit algorithm. The node executing the routines has a neighbourhood of 30 nodes and uses $\eta = 25$ sample points.

Chapter 8

Conclusion

A wireless sensor network represents an evolution of computing that presents some promising opportunities for studying phenomena in large environments over long periods of time. The positioning problem is important issue for sensor networks which arises from deployment techniques that leaves sensor devices at unknown locations in an environment. The use of network positioning with connectivity-based constraints has been identified as a low cost solution to the problem. In many sensor networks the positions of sensors change over time due to physical movement of the phenomena being studied. The continual transformation of the embedding makes the positioning problem especially difficult.

This dissertation studies the positioning problem under a variety of models described in Chapter 2. The set of constraints on a sensor's position defines a

set of feasible positions for a node called the region of uncertainty. The region of uncertainty has been a useful object for the analysis in this work and in particular for thinking about positional error, a central aspect of the positioning problem. This dissertation makes several novel contributions to the body of sensor networks research including a bound on error for applications involving mobile sensor devices, a positioning algorithm for mobile sensor devices called Orbit and the successful implementation and execution of the Orbit algorithm on a sensor network.

The most basic models for communication and mobility were examined in Chapter 4. That chapter defines positional error that is inherent to these models and formulates a lower bound on the minimum achievable positional error of a connectivity-based positioning algorithm. It is notable that this is the first lower bound that applies to mobile sensor scenarios. The model inherent error is a quantification of the slack in the constraints on a sensor's position. The mobility of sensors presents an opportunity for accumulating constraint information over time and that information can be used to reduce positional error. In fact, a common observation of computer simulations of positioning algorithms for mobile systems is that past positional information can be used to improve the most recent position estimate of a sensor. The formulation of the lower bound is the first work to capture the reduction of the size of the region of uncertainty due to the use of past positional information. A subsequent analysis of the lower bound establishes a limit on the

impact on the lower bound due to additional constraint information from the past.

The reduction of the size of the region of uncertainty can result in reduced positional error. In Chapter 6 independent sets of neighbours were identified as a means to impose new constraints on node positions thereby reducing the size of the region of uncertainty. The region of uncertainty becomes smaller with the application of the new constraints but the region can also become disconnected. A disconnected region of uncertainty is problematic for a positioning algorithm. Chapter 6 proposes Orbit, a positioning algorithm that exploits these properties to reduce the positional error of the nodes. The algorithm is designed for mobile sensor networks. A performance evaluation of Orbit demonstrates a significant reduction in positional error when compared with the best algorithm proposed in the literature, WMCL-B.

Though many positioning algorithms have been proposed, few of the algorithms are fully implemented and functional on sensor devices. Chapter 7 describes the successful implementation of Orbit on typical sensor devices which have extreme resource limitations. Despite the limited computational resources available on the devices, the algorithm computes position estimates of each sensor device with errors comparable to computer simulations of the algorithm.

An open area of research is the robust application of negative constraints to the positioning problem. This type of constraint can have a considerable impact on

the reduction of the region of uncertainty. However, the use of negative constraints in practice is problematic. The absence of communication between nodes does not imply the nodes are not within communication range of one another. This problem can exist in both open environments and those with barriers. An environment with barriers also leads to irregular embeddings which can present other challenges for positioning algorithms. This dissertation considered only open environments but a useful extension of this work would examine barriers and the impact they have on negative information.

Bibliography

[1] Isaac Amundson and Xenofon D. Koutsoukos. A survey on localization for mobile wireless sensor networks. In Richard Fuller and Xenofon D. Koutsoukos, editors, *2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, volume 5801 of *Lecture Notes in Computer Science*, pages 235–254. Springer, 2009.

[2] James Aspnes, David Goldenberg, Walter Whiteley, and Brian D. O. Anderson. A theory of network localization. *Transactions on Mobile Computing*, 5(12):1663–1678, 12 2006.

[3] Jonathan Bachrach, Radhika Nagpal, Michael Salib, and Howard Shrobe. Experimental results and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks. *Telecommunications Systems Journal, Special Issue on Wireless System Networks*, 2003.

[4] Aline Baggio and Koen Langendoen. Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 6(5):718–733, 2008.

[5] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks*, 2(2):188–220, 2006.

[6] Pratik Biswas and Yinyu Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN 2004: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 46–54. ACM, 2003.

[7] Barbara A. Block, Heidi Dewar, Charles Farwell, and Eric D. Prince. A new satellite technology for tracking the movements of atlantic bluefin tuna. *Proceedings of the National Academy of Sciences of the United States of America*, 95(16):9384–9389, 1998.

[8] Heinz Breu and David Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry. Theory and Applications*, 9(1-2):3–24, January 1998.

[9] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, 2000.

[10] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Mobile Computing: Special Issue on Mobile Ad Hoc Networking*, 2:483–502, 2002.

[11] Vijay R. Chandrasekhar, Winston Khoon Guan Seah, Yoo Sang Choo, and How Voon Ee. Localization in underwater sensor networks: survey and challenges. In Jun-Hong Cui, Urbashi Mitra, Kevin R. Fall, and Milica Stojanovic, editors, *Underwater Networks*, pages 33–40. ACM, 2006.

[12] Harsha Chenji and Radu Stoleru. Towards accurate mobile sensor network localization in noisy environments. *IEEE Transactions on Mobile Computing*, 99(PrePrints), 2012.

[13] Suprakash Datta, Chris Klinowski, Masoomeh Rudafshani, and Shaker Khaleque. Distributed localization in static and mobile sensor networks. In *WIMOB '06: Proceedings of the 2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 69–76, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer-Verlag, August 2005.

[15] Lance Doherty, Kristofer S. J. Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001: Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1655–1663 vol.3, 2001.

[16] Jacques Dutka. On the problem of random flights. *Archive for History of Exact Sciences Berlin*, 32(3-4):351–375, 1985.

[17] Andreas Frey and Volker Schmidt. Marked point processes in the plane - a survey with applications to spatial modeling of communication networks. *Advances in Performance Analysis*, pages 65–110, 171–214, 1998.

[18] Antonio-Javier Garcia-Sanchez, Felipe Garcia-Sanchez, Fernando Losilla, Pawel Kulakowski, Joan Garcia-Haro, Alejandro Rodríguez, José-Vicente

López-Bao, and Francisco Palomares. Wireless sensor network deployment for monitoring wildlife passages. *Sensors*, 10(8):7236–7262, 2010.

[19] Jack E. Graver. *Counting on Frameworks.* Number 25 in Dolciani Mathematical Expositions. Cambridge University Press, 2001.

[20] Mohamed Hefeeda and Majid Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[21] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *MOBICOM 2004: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 45–57, New York, NY, USA, 2004. ACM.

[22] Barry D. Hughes. *Random walks and random environments.* Oxford University Press, 1996.

[23] Edward V. Huntington. A set of independent postulates for cyclic order. *Proc. Natl. Acad. Sci. USA*, pages 630–631, 1916.

[24] Texas Instruments. MSP430 Wireless Development Tool. `http://www.ti.com/tool/ez430-rf2500`, September 2011.

[25] Bill Jackson and Tibor Jordán. Connected rigidity matroids and unique realizations of graphs. *J. Comb. Theory Ser. B*, 94(1):1–29, 2005.

[26] Xiang Ji and Hongyuan Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *INFOCOM 2004: Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 2652 – 2661, 2004.

[27] Anne-Marie Kermarrec and Guang Tan. Greedy geographic routing in large-scale sensor networks: a minimum network decomposition approach. In *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '10, pages 161–170, New York, NY, USA, 2010. ACM.

[28] Leonard Kleinrock and John Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *NTC '78; National Telecommunications Conference, Volume 1*, volume 1, pages 4.3.2–4.3.5, 1978.

[29] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Network*, 43(4):499–518, 2003.

[30] Mo Li and Yunhao Liu. Rendered path: range-free localization in anisotropic sensor networks with holes. In *MOBICOM 2007: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 51–62, New York, NY, USA, 2007. ACM.

[31] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems Man and Cybernetics Part C: Applications and Reviews*, 37(6):1067–1080, 2007.

[32] Ting Liu, Christopher M. Sadler, Pei Zhang, and Margaret Martonosi. Implementing software on resource-constrained mobile sensors: experiences with impala and zebranet. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 256–269, New York, NY, USA, 2004. ACM.

[33] Juan López, Fulgencio Soto, Andrés Iborra, Pedro Sánchez, and Juan Suardíaz. Design and implementation of a wireless sensor network for precision horticulture. In *Sensor Applications, Experimentation, and Logistics*, 5 2012.

[34] Di Ma, Bang Wang, Hock Beng Lim, and Meng Joo Er. Hop-count based node-to-anchor distance estimation in wireless sensor networks. In *CCNC 2009: Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, pages 920–924, Piscataway, NJ, USA, 2009. IEEE Press.

[35] Stuart MacLean and Suprakash Datta. A lower bound on range-free node localization algorithms. In *ISWCS 2008: Proceedings of the International Symposium on Wireless Communication Systems*, pages 628–632, 2008.

[36] Stuart MacLean and Suprakash Datta. Improving the accuracy of connectivity-based positioning for mobile sensor networks. In *22nd IEEE Personal Indoor Mobile Radio Communications*, pages 1197–1202, 2011.

[37] Stuart MacLean and Suprakash Datta. The minimum positional error incurred by any connectivity-based positioning algorithm for mobile wireless systems. *Theor. Comput. Sci.*, 412(50):6897–6912, 2011.

[38] Yannis Manolopoulos, Dimitrios Katsaros, and Alexis Papadimitriou. Topology control algorithms for wireless sensor networks: a critical survey. In *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, CompSysTech '10, pages 1–10, New York, NY, USA, 2010. ACM.

[39] Madhav V. Marathe, Heinz Breu, Harry B. Hunt, S. S. Ravi, and Daniel J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(2):59–68, 1995.

[40] Hamid Mirebrahim and Mehdi Dehghan. Monte Carlo localization of mobile sensor networks using the position information of neighbor nodes. In *ADHOC-NOW 2009: Proceedings of 8th International Conference on Ad-Hoc, Mobile and Wireless Networks*, pages 270–283, 2009.

[41] Moog Crossbow. XBOW Technology. `http://www.xbow.com`, September 2012.

[42] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In *SenSys 2004: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, 2004. ACM.

[43] Mirco Musolesi and Cecilia Mascolo. *Mobility Models for Systems Evaluation*, page 43. Springer-Verlag, 1st edition, 2009.

[44] Radhika Nagpal, Howard Shrobe, and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN 2003: Proceedings of the 2nd international conference on Information processing in sensor networks*, pages 333–348, Berlin, Heidelberg, 2003. Springer-Verlag.

[45] William Navidi and Tracy Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3:99–108, January 2004.

[46] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS). In *GLOBECOM 2001: Proceedings of the IEEE conference on Global telecommunications*, volume 5, pages 2926 –2931, 2001.

[47] Dragos Niculescu and Badri Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22(1-4):267–280, 2003.

[48] Regina O'Dell and Roger Wattenhofer. Theoretical aspects of connectivity-based multi-hop positioning. *Theor. Comput. Sci.*, 344(1):47–68, 2005.

[49] Chulsung Park. EcoMote.net. `http://www.ecomote.net/`, September 2012.

[50] Chulsung Park, Jinfeng Liu, and Pai H. Chou. Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring. In *IPSN '05:*

Proceedings of the 4th international symposium on Information processing in sensor networks, page 54, Piscataway, NJ, USA, 2005. IEEE Press.

[51] Stephen C. Peterson and Peter B. Noble. A two-dimensional random-walk analysis of human granulocyte movement. *Biophysical Journal*, 12(8):1048–1055, 1972.

[52] C.R. Rao, J.P. Miller, and D.C. Rao, editors. *Handbook of Statistics: Epidemiology and Medical Statistics*, volume 27. Elsevier, 2008.

[53] Masoomeh Rudafshani and Suprakash Datta. Localization in wireless sensor networks. In *IPSN 2007: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 51–60, New York, NY, USA, 2007. ACM.

[54] Nordic Semiconductor. nRF24LE1: 2.4GHz RF System-on-Chip with Flash. `http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24LE1`, October 2011.

[55] Richard Serfozo. *Basics of Applied Stochastic Processes.* Probability and Its Applications. Springer-Verlag, January 2009.

[56] Yi Shang and Wheeler Ruml. Improved MDS-based localization. In *INFOCOM 2004: Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2640 – 2651, 7-11 2004.

[57] Yi Shang, Wheeler Ruml, and Markus P.J. Fromherz. Positioning using local maps. *Ad Hoc Networks*, 4(2):240 – 253, 2006.

[58] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz. Localization from mere connectivity. In *MOBIHOC 2003: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 201–212, New York, NY, USA, 2003. ACM.

[59] Michel Sortais, Sven D. Hermann, and Adam Wolisz. Analytical investigation of intersection based range-free localization. *Annals of Telecommunications*, 63(5):307–320, 06 2008.

[60] Ivan Stojmenovic, editor. *Handbook of wireless networks and mobile computing.* John Wiley & Sons, Inc., New York, NY, USA, 2002.

[61] Gideon Stupp and Moshe Sidi. The expected uncertainty of range-free localization protocols in sensor networks. *Theoretical Computer Science*, 344(1):86 – 99, 2005. Algorithmic Aspects of Wireless Sensor Networks.

[62] Eric Weisstein. *CRC Concise Encyclopedia of Mathematics.* CRC Press, 1999. pp. 249-250.

[63] Wei Xi, Yuan He, Yunhao Liu, Jizhong Zhao, Lufeng Mo, Zheng Yang, Jiliang Wang, and Xiangyang Li. Locating sensors in the wild: pursuit of ranging quality. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 295–308, New York, NY, USA, 2010. ACM.

[64] Wei Xi, Jizhong Zhao, Xue Liu, Xiang-Yang Li, and Yong Qi. EUL: An efficient and universal localization method for wireless sensor network. In *Proceedings of the International Conference on Distributed Computing Systems*, pages 433–440, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[65] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. In *INFOCOM 2003: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1312–1321. IEEE, April 2003.

[66] Shigeng Zhang, Jiannong Cao, Lijun Chen, and Daoxu Chen. Locating nodes in mobile sensor networks more accurately and faster. In *SECON 2008: Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 37–45, 2008.

[67] Shigeng Zhang, Jiannong Cao, Lijun Chen, and Daoxu Chen. Accurate and energy-efficient range-free localization for mobile sensor networks. *IEEE Transactions on Mobile Computing*, 9:897–910, 2010.

[68] Ziguo Zhong and Tian He. Achieving range-free localization beyond connectivity. In *SenSys 2009: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 281–294, New York, NY, USA, 2009. ACM.

[69] Reinholds Zviedris, Atis Elsts, Girts Strazdins, Artis Mednis, and Leo Selavo. Lynxnet: wild animal monitoring using sensor networks. In *Proceedings of the 4th international conference on Real-world wireless sensor networks*, REALWSN'10, pages 170–173, Berlin, Heidelberg, 2010. Springer-Verlag.